Tepper's Navigator: A 24/7 Chatbot Companion

Linguabots: Kanishka Soni, Ning Ren, Sherine George, Amirthavalli Narayanan,

Xiaohuan Wang, and Wendy Kuo

Tepper School of Business, Carnegie Mellon University

46-992 Experiential Learning

Professor Ganesh Mani

December 8, 2023

Table of Contents

Overview **3**

Problems identified in the previous solution **5**

Proposed Solution **7**

Challenges Encountered **11**

Future Prospects **12**

Conclusion **13**

References **14**

Appendix A: Sample Training Data **15**

Appendix B: Additional Resources **16**

Overview

Mini 1: AI HelpBot Development with GPT-3.5 API and Flask

In Mini 1, we achieved a significant milestone by developing an AI chatbot using the GPT-3.5 API. Our primary focus was on the real-time delivery of information. We created a simple user interface using Flask, making it easy for Tepper School of Business's MSBA students to interact with the chatbot. We also implemented a continuous monitoring and updating system to keep our HelpBot current with the latest information, reflecting the ever-changing academic environment. User feedback integration was integral to our approach. This allowed us to iteratively refine our search algorithms and relevance ranking models based on user input, improving the HelpBot's efficacy over time.

Mini 2 Hackathon: Transition to PALM2 API and Streamlit Dashboard

In Mini 2, we enhanced our HelpBot by transitioning to the **PALM2 API**, a more advanced language model. We also adopted **Streamlit** to create a user-friendly dashboard interface, further improving the user experience. To make the HelpBot even more versatile, we leveraged **Open Router technology**, enabling us to connect to a range of Language Models (LLMs). This allowed us to perform comparative analyses and select the most suitable LLM for specific tasks, ensuring the highest quality responses and information retrieval. We added valuable features like assignment structuring and text summarization to the HelpBot, simplifying students' academic tasks and making it easier for them to navigate coursework and manage assignments.

In summary, Mini 1 marked the initial development of our AI chatbot using the GPT-3.5 API, focusing on real-time information delivery and user feedback integration. In Mini 2, we upgraded to the PALM2 API, created a user-friendly Streamlit dashboard, and integrated open

router technology for comparative LLM analysis. Additionally, we added features like assignment structuring and text summarization, enhancing the HelpBot's utility for Tepper School of Business's MSBA students. However, during the Mini 2 testing and development phase, we encountered several challenges that highlighted areas for improvement in our chatbot. The primary issues identified were:

1. **Relatively Slow Results:** The chatbot's response time, often extending to 15 seconds, affected the user experience, underscoring the need for speed in information retrieval.
2. **Complexity in Generating User Prompts:** We faced challenges in developing an exhaustive set of user prompts that adequately covered the range of potential queries, which was crucial for ensuring the precision and quality of responses.
3. **Issues of Accuracy:** The chatbot occasionally provided inaccurate or irrelevant responses to queries that were phrased differently from its training data, highlighting a critical need for adaptability and accuracy in varying contexts.

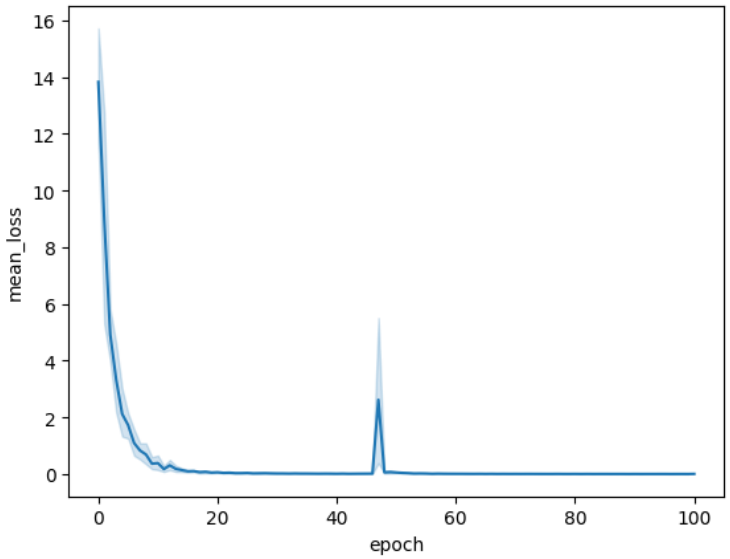

Fig 1: Loss curve to show deviation between actual and predicted values

To overcome these challenges, we proposed a solution involving the integration of *LangChain, and FAISS.* This approach aimed to refine data acquisition and preprocessing, enhance semantic similarity search for accurate information retrieval, and implement a multi-layered strategy for relevance ranking and filtering. Additionally, we prioritized the real-time delivery of information and the integration of user feedback mechanisms, ensuring the chatbot continuously evolves and improves based on user interactions. This comprehensive strategy is anticipated to markedly boost the chatbot's efficiency and user-centric approach, enabling it to adeptly navigate extensive data repositories and promptly deliver accurate, relevant information. This evolution from the initial development in Mini 1 to the refinement and enhancement in Mini 2 illustrates a commitment to ongoing improvement and adaptation, reflecting our dedication to providing the Tepper community with an effective, user-friendly information management tool.

Problems identified in the previous solution

Relatively Slow Results

The evaluation and continual monitoring of LLMs invariably prioritize response time as a foundational metric. The efficacy of these systems heavily relies on prompt replies to user queries. However, our initial iteration of the HelpBot encountered a notable setback in response time, often necessitating 15 seconds to generate user responses. This sluggishness starkly contrasts the near-instantaneous responsiveness observed in systems like ChatGPT, leading to a significant deterioration in user experience. Consequently, users might opt to disengage from or abandon the HelpBot without obtaining the required response.

The challenge of sifting through extensive data repositories across various websites poses a hurdle for systems in pinpointing precise, pertinent information. In the case of the HelpBot,

designed to offer immediate, pragmatic aid to students, this struggle to discern accuracy and relevance likely contributes to its diminished effectiveness. Consequently, this impediment undermines users' ongoing engagement and substantially hampers the system's real-time utility, potentially contributing to the observed slow response time.

Moreover, this delayed response undermines the seamless retrieval of information, directly contradicting the HelpBot's primary objective: to enhance efficiency among Tepper students by furnishing timely academic information related to tasks and assignments. As a result, the protracted response time not only frustrates users but also compromises the original intent and utility of the HelpBot.

Complexity in generating user prompts

Prompt engineering stands as a pivotal aspect in tailoring an LLM for specific use cases, such as our instance where the LLM was trained to address queries specific to the MSBA course at Tepper School of Business. This training process revolves around furnishing the LLM with a comprehensive array of user prompts and their corresponding responses. The primary challenge in this endeavor lies in meticulously curating an exhaustive catalog of LLM prompts that cover a broad spectrum of potential questions the LLM might encounter.

For instance, while training the LLM to respond to inquiries like "Who teaches the course Experiential Learning 2?" allows the model to autonomously comprehend and address similar questions with relevant data, it necessitates prior training to handle variations such as "Who is the TA for the course Experiential Learning 2?". Without such specific training, the model tends to provide inaccurate or incomplete responses. The intricacy involved in devising these user prompts directly impacts the precision and quality of responses, thereby diminishing the overall effectiveness of the HelpBot.

Issues of Accuracy

In our evaluation using Google AI PaLM 2, an observed behavior revealed the model's inclination to scour the web for information, even if it is irrelevant, when faced with a differently worded query from its training data, despite conveying a similar intent. For instance, in our initial HelpBot version, when queried about Kevin Dietrick, the model frequently included data pertaining to an unrelated individual found on the internet, resulting in erroneous responses.

The sporadic provision of inaccurate or irrelevant information poses a threat to the reliability of HelpBot, instigating doubts among users regarding the accuracy of the information provided. This decline in reliability not only erodes user confidence in HelpBot but also introduces uncertainty among users, potentially influencing their decision-making process based on the information received.

Proposed Solution

In addressing the challenge of navigating vast data repositories and extracting precise, pertinent information, the fusion of LangChain and FAISS emerges as a robust solution. The methodology adopted underscores a multi-faceted approach harnessing the strengths of both technologies.

LangChain and FAISS Overview

LangChain: LangChain is a framework for developing applications powered by language models. It provides modular abstractions for the components necessary to work with LLMs while also leveraging the reasoning capabilities of LLMs to perform tasks.

FAISS: FAISS, or Facebook AI Similarity Search, is a library that unlocks the power of similarity search algorithms, enabling swift and efficient retrieval of relevant documents based

on semantic similarities. Its high-dimensional indexing capabilities and fast search performance become our compass, directing us towards the most pertinent documents it stores as vectors.

Data Acquisition and Preprocessing

Our initiative commenced with data acquisition and preprocessing. We harnessed LangChain's adept web scraping capabilities to aggregate pertinent data from diverse online sources. This data underwent meticulous cleansing and standardization through LangChain's Natural Language Processing (NLP) tools, systematically eliminating noise and inconsistencies, ensuring our subsequent analysis rested on refined information.

Although data preparation was a comparatively difficult task we managed to divide it into four segments and synthesize it. We seamlessly segmented the process into four key categories: *Academic, Faculty, Admissions, and Lifestyle.* At present, we've harnessed our capabilities to train the model effectively on Academic and Faculty data. Our vision is to extend this by incorporating the remaining segments in the near future. For a detailed view of the sample training data used in the development of the chatbot, please refer to *Appendix A*.

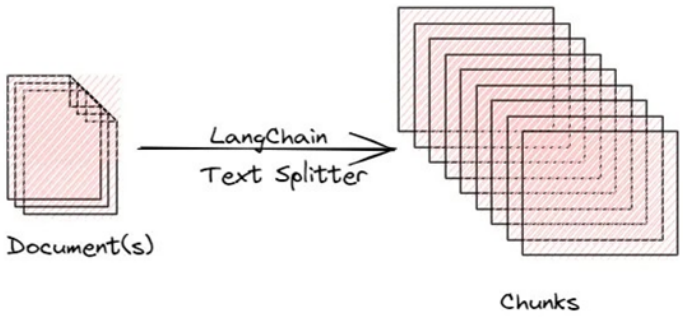

Fig 2: FidConverting Chunks into Embeddings and Storage

Following data acquisition and preprocessing, we convert document chunks into embeddings using LangChain's advanced models. These embeddings are then stored in a vector database, forming the core of our information retrieval system. This enables rapid access and retrieval of relevant data, ensuring swift responses to user queries, often within just 15 seconds, while maintaining accuracy and relevance.

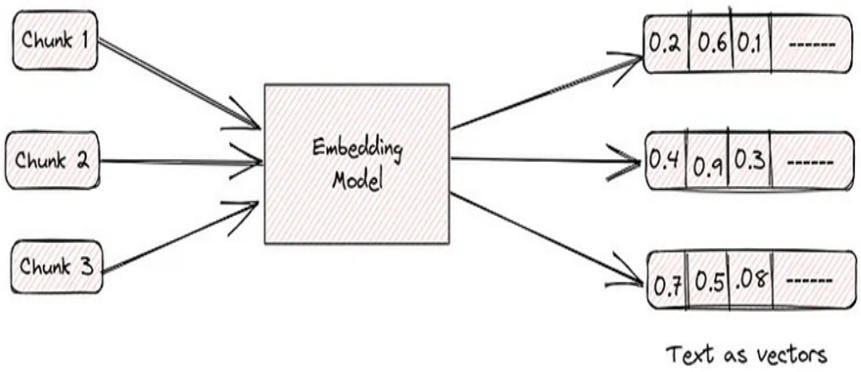

Fig 3: Convert each document chunk into embedding

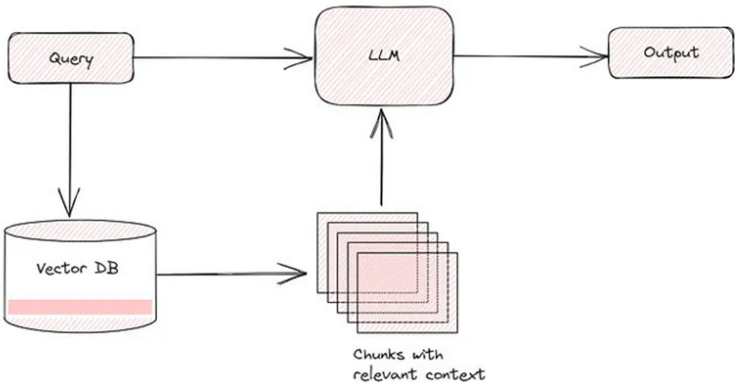

Fig 4: Storing the embeddings in a Vector DB

Semantic Similarity Search with FAISS

The crux of our strategy lies in the semantic similarity search powered by FAISS. We utilized LangChain's embedding models to extract meaningful representations from the data, enabling FAISS's efficient similarity search algorithms to retrieve documents aligned semantically with user queries. This seamless integration substantially optimized our search process, significantly enhancing the precision of information retrieval.

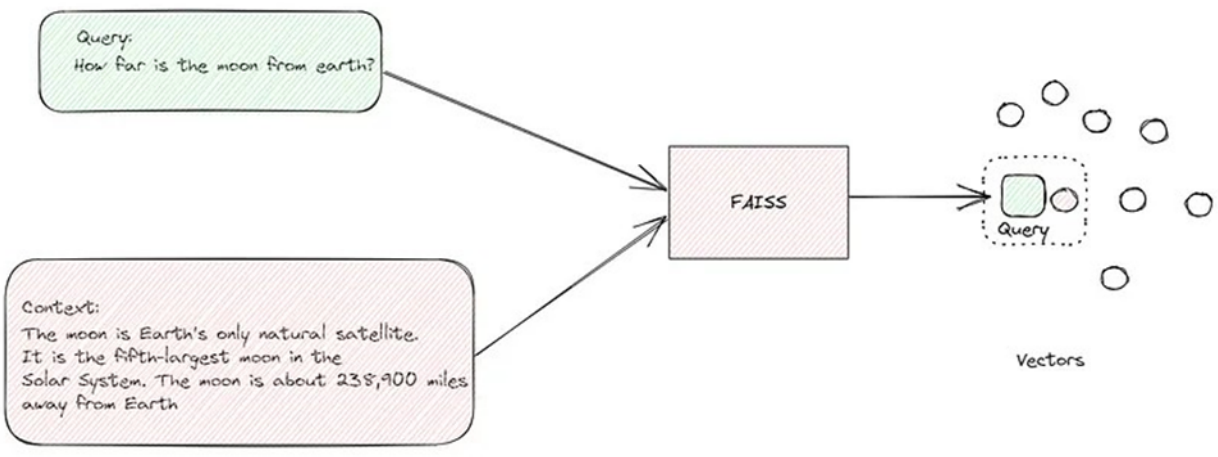

Fig 5: FAISS similarity search example

Relevance Ranking and Filtering

Furthermore, we implemented a multi-layered approach involving relevance ranking and filtering. Leveraging LangChain's machine learning capabilities, we devised a nuanced relevance ranking mechanism, considering variables like query relevance, source credibility, and user preferences. Simultaneously, our system efficiently filtered out redundant or irrelevant documents, ensuring users received only the most pertinent information.

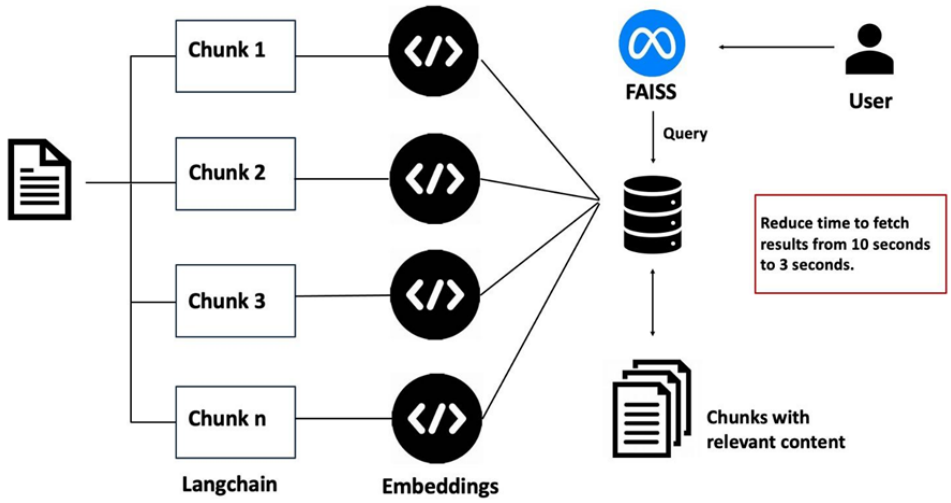

Fig 6: High-level design

In summary, our amalgamation of LangChain and FAISS, along with a comprehensive approach encompassing data acquisition, semantic search, relevance ranking, real-time delivery, and user feedback integration, fortified our HelpBot's ability to navigate extensive data repositories and promptly provide users with accurate, relevant information. This holistic strategy marked an evolution towards an increasingly effective and user-centric information retrieval system.

Challenges Encountered

1. **Data Quality and Quantity:** During the process of training data, it's impossible to cover all the potential queries of users. Data sometimes may not be updated on time to reflect the most current information of the programme.
2. **Natural Language Understanding:** There exist challenges in accurately interpreting the demands of the users. Sometimes, it's difficult to extract the key information from complex queries. The system has an inadequate ability to handle nuanced language and contexts.

3. **Integration with Existing Platforms:** Difficulty in integrating the HelpBot with other platforms of Tepper. There exists the issue of incompatibility with other systems and interfaces.
4. **Performance and Scalability:** The User experience has been negatively impacted by the slow response times and there will become the potential challenges in handling the growing user base and increased volume of queries.

Future Prospects

In envisioning the future development of the chatbot, several key areas merit attention. Personalization and Customization should be prioritized, enhancing the chatbot's ability to tailor responses based on user preferences, academic history, and specific needs. Simultaneously, extending Multilingual Support is essential to accommodate the diverse international student body commonly found in a university setting like CMU. Integration with Campus Services remains a pivotal goal, necessitating collaboration with various departments such as library services, IT support, student services, and CAPS to seamlessly integrate the chatbot with their systems. Real-time Assistance and Updates are crucial for keeping students and staff informed about campus events, class changes, or urgent notifications. Performance Metrics and Analytics must be continually refined, encompassing response accuracy, user satisfaction, and engagement rates, with ongoing user feedback informing iterative improvements. Finally, prioritizing Security and Data Privacy is imperative, requiring the implementation of robust measures to protect user data and ensure compliance with privacy regulations. This comprehensive approach to future development ensures the chatbot's evolution aligns with the dynamic needs of the university community.

Conclusion

As we conclude the Tepper's Navigator project, our journey in developing and refining this 24/7 chatbot companion offers significant insights. Starting with the initial trials using the GPT-3.5 API and evolving through the integration of advanced technologies like LangChain and FAISS, our experience highlights both the potential and the challenges in the realm of AI-driven solutions.

The transition from basic interaction models to more sophisticated systems demonstrates our commitment to improving user experience and operational efficiency. However, it also underscores the complexities inherent in AI applications, such as the need for speed, accuracy, and versatility in handling diverse user queries.

Our challenges, particularly in response time, prompt generation, and maintaining accuracy, have not only tested our capabilities but also pushed us to innovate and adapt. These hurdles have been crucial in steering the project towards more effective and reliable solutions, emphasizing the importance of continuous learning and improvement in technology projects.

Looking forward, the lessons learned from Tepper's Navigator will inform future projects. The balance between technological advancement and practical usability will remain a key focus. As we progress, we aim to leverage these experiences to develop solutions that are not only technologically sound but also deeply aligned with user needs and expectations.

In conclusion, Tepper's Navigator stands as a significant step in our journey towards integrating AI into daily academic and administrative processes. It represents a blend of ambition, technical expertise, and a deep understanding of user requirements. As we wrap up this project, we are excited about the potential of such technologies to transform educational environments and enhance the overall campus experience.

References

API documentation :   palm API :   Generative AI for developers. Generative AI for Developers. (n.d.). https://developers.generativeai.google/api

Langchain. (n.d.). https://python.langchain.com/docs/get_started/introduction

Mohiuddin, A. (2023, August 9). Using AI to chat with documents: Leveraging Langchain, FAISS, and Openai. Medium. https://medium.com/@ahmed.mohiuddin.architecture/using-ai-to-chat-with-your-documents-leveraging-langchain-faiss-and-openai-3281acfcc4e9

Streamlit. (n.d.). https://streamlit.io/

Welcome to FAISS documentation. Welcome to Faiss Documentation - Faiss documentation. (n.d.). https://faiss.ai/index.html

Appendix A: Sample Training Data

Q. Who is the executive director for the Master Student Services?

A. Wendy Hermann is the executive director for the Master Student Services.

Q. Who is teaching Experiential Learning?

A. Ganesh Mani will be teaching Experiential Learning in mini 3.

Q. What is covered in the Optimization for Prescriptive Analytics course?

A. In this course, the framework and techniques of mathematical optimization are introduced. The course emphasizes modeling. That is, creating a mathematical description that reflects a given real-world problem. Equipped with these models, we then discuss the necessary mathematical techniques for finding optimal solutions. Finally, we consider the solution to these problems using optimization software, i.e. we represent the models in either Excel or Python and use a solver to compute an optimal solution.

Appendix B: Additional Resources

ChatGPT

[1] https://chat.openai.com/share/ae97821b-524f-418e-9b2b-6e769af9666f

[2] https://chat.openai.com/share/cd11100b-5370-42c2-8dee-3861ae66b814

GitHub URL

https://github.com/sherinegeorge21/Hackathon

Demo Video

https://youtu.be/DnathemT5vs