Language Models Training Humans

Yusuf Anı
Technical University of Munich, Germany
yusufani8@gmail.com

Abstract

This study delves into the transformative potential of Large Language Models (LLMs) in reshaping learning paradigms. At the heart of our investigation is the deployment of LLM. Our research introduces a personalized agent that is meticulously designed to leverage the capabilities of LLMs to offer tailored learning experiences. By integrating the Langchain library with GPT-4 Turbo, we've crafted an architecture that enhances learning through personalized curriculums. This agent promises an immersive learning journey marked by personalized content, Q&A sessions, and comprehensive evaluations, all fine-tuned to the learner's preferences and skill level. Our work paves the way for a future where learning is not just personalized but deeply interconnected with the advancements in artificial intelligence.

1 Introduction

Learning has continually evolved across the span of human history, and the requirement of learning always remained indispensable. In the modern era, the advent of Large Language Models (LLMs), with their burgeoning popularity in the field of artificial intelligence, poses a compelling question: Can the process of learning be tailored and expedited? LLMs, sophisticated AI systems adept at interpreting and fabricating text for coherent interaction, have revolutionized tasks in natural language processing, achieving near-human proficiency across various domains. These models stand out for their emergent capabilities, including reasoning, planning, decision-making, and contextual learning, attributes fostered by their extensive scale. Their application has been vast, from multimodal interfaces to robotics and autonomous entities, underscoring their transformative impact on technology. Drawing inspiration from these developments, we have designed a personalized agent aiming to harness the potential of LLMs for customized learning experiences.

There are many well-known LLMs, such as the GPT models from OpenAI(ope, 2024), LLama models from Meta(Touvron et al., 2023), as well as Gemini(Team et al., 2023) and Gemma(Gemma Team et al., 2024) from Google, Claude models from Anthropic(Cla, 2024) and Mistral(Jiang et al., 2023). Although each has its strengths in generating relevant content, we opted for GPT-4 Turbo Our decision was influenced by benchmark results that highlighted GPT-4 Turbo as the superior choice:

- LMSYS Chatbot Arena, a crowdsourced platform for evaluating LLMs, has amassed over 300,000 human preference votes, employing the Elo ranking system to rank them. GPT-4 Turbo leads with an Elo score of 1251(Zheng et al., 2023)
- The MT-Bench score evaluates chatbots on open-ended questions, focusing on their multi-turn conversational and instruction-following capabilities. Scores are derived from human ratings through expert or crowdsourced votes. A higher MT-Bench score indicates closer alignment with human preferences. GPT-4 Turbo excels with a score of 9.32.(Zheng et al., 2023)
- AlpacaEval, an LLM-based evaluation, is fast, affordable, and reliable, utilizing the AlpacaFarm set to assess models' ability to follow general user instructions. It compares responses to reference responses, with GPT-4 Turbo achieving a 50.00% win rate, indicating high concordance with human annotations and reflecting its leadership in AlpacaEval's rankings.(Dubois et al., 2023)

We have integrated the Langchain library(Chase, 2022) with the GPT-4 Turbo LLM to enhance our setup. Langchain has enabled us to craft our architecture, leveraging agents and chains with most

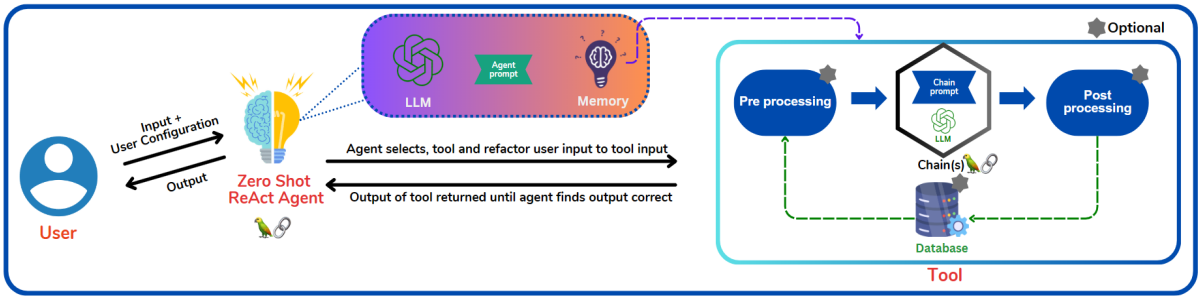

Figure 1: Overview of our method's architecture. It is built on a Langchain agent and a suite of tools. The agent determines the appropriate tool to take action, guided by the user's query and configuration. Each tool operates within a chain, incorporating pre-processing and/or post-processing code as needed. Every action taken is meticulously logged in a database and saved in memory.

of the LLM support. We have developed an architecture that transforms the learning experience by setting prompts with Python commands within our chains. It delivers a personalized curriculum, personalized content and feedback, question-and-answer sessions, tests, flashcards, and reports, all designed to cater to the individual learner's needs. We have selected the cooking domain to limit our research topics, but our algorithm can easily be generalized to use it everywhere.

2 Related Work

Many apps have been crafted by the hands of LLM agents and their intricate chains. Yet, only a handful truly delve into the realm of personalized teaching. Mr. Ranedeer AI Tutor(mr, 2024) stands out in this niche, offering a tailored teaching experience with customizable settings such as tone, depth of understanding, use of emojis, communication style, and a reasoning engine based on the OpenAI platform. Despite its promising potential, this project is still in the throes of development. A critical reliance on OpenAI agents, which require premium access and are exclusively built on the GPT-4 model, marks a notable limitation. In contrast, our work distinguishes itself by compatibility with any LLM of choice and a robust feedback system through tests and reports, as mentioned in the method section.

3 Method

Agents in LangChain are systems that use a language model to interact with other tools. They can be used for tasks such as grounded question/answering, interacting with APIs, or taking action. We used Zero shot reAct Agent which performs a reasoning step before acting. It's suitable for scenarios where an immediate response is required without prior training. The agent is capable of remember what kind of action are taken so far. It decides new action based on user configuration, user input, memory and the prompt where we give all directions to agent. Generally this prompt includes duty of agent, what kind of tools it has, and important points while taking action.

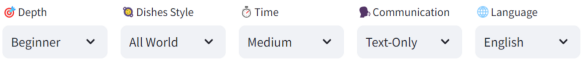

Figure 2: Screenshot of user configuration where user can change any time in the learning process

Listing 1: Simpified agent prompt for the personalized cooking assistant

```
You are a personalized cooking assistant
that aims to help the user cook and
answer their question about cooking. If
the user wants to learn how to cook a
dish, you try your best to follow the
user's configuration and provide a
suitable and perfect curriculum for
them to teach how to cook this dish.
You also be able to answer their
questions, you try to answer in as
detailed as you can, and make it
easy to understand. When you generated
curriculum, you must wait for user
input to proceed (start teaching)
any module.
```

The architecture of our projects relies on Langchain agent, tools, and chains as in figure 1. At the heart of LangChain are its agents – dynamic systems leveraging language models to seamlessly interface with a variety of tools. These agents are versatile, handling tasks from grounded question-answering to API interactions and proactive actions. A standout in our toolkit is the Zero-shot reAct Agent, designed for reasoning before taking action. This agent thrives in environments demand-

ing swift, informed responses without the need for preliminary training. It decides new actions based on user configuration, user input, memory, and the prompt as in listing 1 where we give all directions to the agent. Guiding these decisions is a prompt that outlines the agent's responsibilities, available tools, and critical considerations for action. This prompt acts as a comprehensive blueprint, ensuring actions are both relevant and impactful. The agent is also responsible for refactoring user input to the tool's expected input.

User configuration is illustrated in figure 2, where the user can select their preferences even after initiating the conversation. Even if we have significantly benefited from Mr.Ranedeer's project options, the user can add any additional features to the system. The built-in configurable options include:

Figure 3: Avaliable tools in our work

- **Depth Option:** Allows users to select their cooking expertise level, offering choices of Beginner, Intermediate, and Expert. This customizes the content to match their skill level.
- **Style Option:** Provides a selection of cuisines from across the globe, including Asian, European, American, South American, and African, as well as an "All World" option for those open to any culinary style.
- **Time Option:** Enables users to specify the amount of time they wish to spend on cooking, with options ranging from Short, Medium, to Long. This affects the complexity and duration of the recipes presented by adjusting a number of modules generated.
- **Communication Option:** Determines the format of the content, choosing between Image-Containing and Text-Only, to cater to the user's preference for visual or textual guidance.
- **Language Option:** Offers the application in multiple languages, including English, Chinese, Turkish, and German, ensuring users can engage in their preferred language.

The application monitors changes in these settings to tailor the content accordingly, highlighting a personalized and adaptable user experience. All user configurations are directly incorporated into the tool's prompt, significantly influencing the results.

Langchain tools are interfaces that an agent can use to interact with the world. Each tool consists of Chain and pre-processing or post-processing or both of them. Langchain Chains refer to sequences of calls - whether to an LLM, a tool, or a data preprocessing step. We used Chain to generate user content. We also used codes to refactor tool input and output and interact with the user interface. We will explain each module in our project in the following sections, as illustrated in Figure 4.

3.1 Curriculum Tool

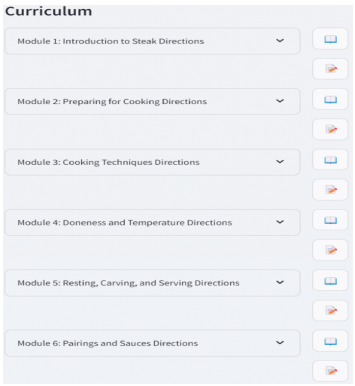

Figure 4: Example curriculum on How to cook steak with Long time option selected

The Curriculum Tool is pivotal in crafting a structured learning journey. By generating a curriculum based on a specific input, it breaks down the overarching topic into manageable modules, each separated by a unique delimiter for easy parsing. This tool is crucial when initiating a new learning path, ensuring that the content is both organized and tailored to the user's preferences. It underscores the importance of structured learning, guiding users through a coherent sequence of topics and reinforcing knowledge acquisition through well-defined stages.

3.2 Chat Tool

The Chat Tool stands as the system's communicative backbone, facilitating straightforward interactions between the user and the system. It operates on a direct input-output basis, echoing the user's queries or statements. This tool comes into play when the interaction doesn't align with the predefined tasks of other tools, serving as a catch-all for user inputs that require a simple, conversational response. Its importance lies in maintaining engagement and fluidity in user interactions, ensuring that all user inputs receive acknowledgment, thus fostering a more human-like conversation flow.

3.3 QA Tool

Embedded within the system's fabric, the QA Tool harnesses the power of the langchain to provide detailed answers to user queries. This feature is particularly beneficial in educational contexts, where users might have specific questions related to the curriculum content. By drawing from the system's knowledge base and the context of previous interactions, it delivers precise, context-aware responses. The QA Tool is indispensable for enriching the learning experience, offering instant clarifications and deepening understanding on demand.

3.4 Teach Tool

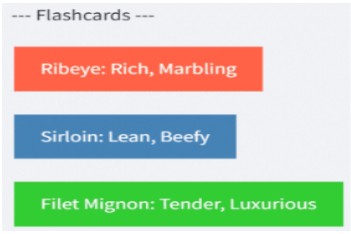

Figure 5: Example flashcards for How to cook steak topic

The Teach Tool is designed to deliver educational content within the curriculum's framework, adapting to the nuances of user-configured preferences and external configurations. It has three primary responsibilities: first one dynamically generating content for a specified module, integrating additional instructions as necessary. Secondly, it created flashcards for the user to point out essential concepts in selected module like in figure 5. Lastly, if the user selects a visual learning style, it generates the main image for the topic by Dalle-3 using OpenAI APIs.

This tool is crucial when advancing through the curriculum, ensuring that each module is both informative and customized to the learner's needs. It underscores the system's adaptability and its commitment to providing a personalized learning experience by features like image generation and flashcards.

3.5 Evaluate Tool

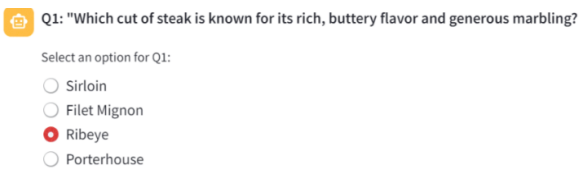

Figure 6: Example question for How to cook steak topic

The Evaluate Tool is instrumental in assessing the user's comprehension and retention of module content. It crafts quizzes and exercises that test the user's knowledge, generating them based on the specific content of each module. This tool is essential for the iterative learning process, offering a tangible measure of the user's progress and understanding. Through evaluation, it closes the feedback loop, enabling users and the system alike to gauge the effectiveness of the learning experience.

3.6 Analyze Tool

The Analyze Tool undertakes a comprehensive review of user performance, particularly after quiz attempts. It analyzes quiz results to offer a detailed breakdown of strengths and areas for improvement. This tool is invaluable for reflective learning, allowing users to understand their progress, rectify misunderstandings, and reinforce knowledge. By providing targeted feedback, it plays a crucial role in fostering a deeper, more meaningful learning journey.

4 User Interface

We have used Streamlit (str, 2024) for the user interface. Streamlit is an open-source Python framework for machine learning and data science teams. It can create dynamic websites in a very short time. It also supports LLM applications where used chains are displayed automatically. It also gives the ability to store user variables in session states. We have used this feature as a database in our design.

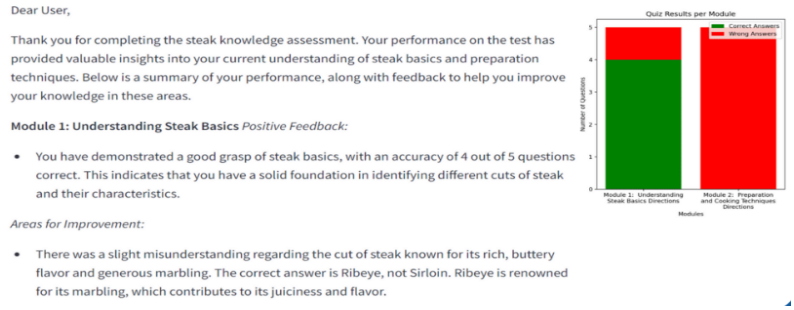

Figure 7: Example analyze for How to cook steak topic where the user only answered two modules' quizes

5 Discussion and Limitations

The exploration into LLMs has unfolded avenues where personalized learning experiences could be vastly improved. Our study leveraged GPT-4 Turbo for its exceptional performance, as illustrated by its leading scores in LMSYS Chatbot Arena, MT-Bench, and AlpacaEval. This choice was strategic, aiming to harness its superior conversational and instruction-following capabilities for educational purposes. However, no study is without its challenges and limitations, which we candidly acknowledge and address herein.

Firstly, the performance testing of our study leans heavily on human annotations—a necessity, yet a limitation. The inherent subjectivity and variability in human judgment can introduce biases or inconsistencies in performance evaluation. However, we propose a mitigating solution through our Analyze Tool, which provides a quantifiable measure of accuracy. This, paired with detailed analysis, carves a pathway to attain a more objective evaluation framework, albeit with room for further refinement.

A notable concern in the utilization of LLMs, including GPT-4 Turbo, is the phenomenon of hallucination. These models, despite their intelligence, occasionally generate information that is factually incorrect or irrelevant—a drawback that necessitates vigilant monitoring and correction in educational settings.

Additionally, the process of generating content with LLMs, particularly complex and personalized learning material, is time-intensive. This delay could impact the immediacy of learning experiences, potentially hindering engagement or the flow of interactive sessions.

Lastly, we acknowledge the constraint imposed by GPT-4 Turbo's limited context length of 128k. While this capacity suffices for most short courses, it poses a challenge for more extensive learning materials. This limitation underscores the necessity for strategic content segmentation or the exploration of alternative models for longer courses.

6 Conclusion

In conclusion, our investigation into the potential of LLMs for personalized learning experiences has underscored both the remarkable capabilities and the inherent limitations of these models. GPT-4 Turbo, with its robust performance and adaptability, presents a promising avenue for educational applications. Our study has made strides in crafting a personalized agent capable of delivering customized learning experiences, yet it also highlights the critical need for ongoing research and development. The limitations of human annotation dependence, model hallucinations, content generation time, and context length restrictions present avenues for further innovation. Addressing these challenges will be crucial in refining the efficacy and reliability of LLM-based educational tools, paving the way for a future where personalized learning is not just a vision but a practical reality.

7 Acknowledgements

We would like to thank Tobias Eder for his guidance and supervision during the project. Additionally, we would like to thank the Social Computing Chair at TUM, for the provided OpenAI credits.

References

2024. Claude 2 anthropic. https://www.anthropic.com/news/claude-2. Accessed: 2024-03-08.

2024. Jushbjj/mr.-ranedeer-ai-tutor: A gpt-4 ai tutor prompt for customizable personalized learning experiences. https://github.com/JushBJJ/Mr.-Ranedeer-AI-Tutor. Accessed: 2024-03-10.

2024. New embedding models and api updates. https://openai.com/blog/new-embedding-models-and-api-updates. Accessed: 2024-03-08.

2024. Streamlit • a faster way to build and share data apps. https://streamlit.io/. Accessed: 2024-03-10.

Harrison Chase. 2022. LangChain.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback.

Thomas Mesnard Gemma Team, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, and et al. 2024. Gemma.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Gemini Team et al. 2023. Gemini: A family of highly capable multimodal models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.