AI Therapist

Junyao Shangguan, Jingyuan Shi

Introduction

For a long time, people have been curious about whether machines can understand how we feel. One of the first tries was with a program called ELIZA in 1966, which could chat in a way that made it seem like it understood people more than it really did. Nowadays, we have much smarter computer programs, known as Large Language Models (LLMs), which are even better at understanding and responding to our emotions. This project aims to create a chatbot that can offer mental health support, kind of like a therapist, but powered by AI.

The reason we're doing this is simple: many people struggle with mental health issues, but not everyone can or wants to go see a therapist. So, we thought, why not make an AI therapist? This way, people can get help privately and easily, without having to worry about the stigma or finding a human therapist they can afford and trust.

By comparing how this AI therapist talks and helps people with how human therapists do, we can learn a lot. This includes understanding more about what makes a good therapist and how AI can fit into the world of mental health care. It's not just about making something cool with technology; it's about helping people feel better and exploring how AI can do good in a very human part of our lives.

Data

For our project, we used the Counsel Chat dataset. This dataset comes from a website called CounselChat.com, where therapists help people by answering their questions. The dataset is rich with answers from these experts and covers many different topics about mental health.

You can find the original dataset here: Hugging Face. It includes 2.78k entries, each with a title, a question, an answer, and a number of upvotes. All these are single-turn conversations about mental health. To make sure our training dataset was balanced, we chose the most upvoted answer for each question, giving us about 900 unique pairs of questions and answers.

To make our model better at handling dialogue, we designed a pipeline prompting GPT-4 to create new conversations that look like real dialogues from single-turn conversations. We got about 900 dialogues, each about 1000 tokens long. At last, we formatted these dialogues to the Llama 2 chat template, which is designed to do self-supervised training for Llama. We released our dataset here.

Model

In our project, we used two main types of models. The first type includes online models accessed through the OpenAI API, specifically GPT-3.5 and GPT-4. The second type consists of local models, including LLaMA 2 (7B), LLaMA 3 (8B), Mistral (7B), and our own fine-tuned LLaMA model.

We ran the local models using a tool called Ollama. Ollama is an advanced AI tool that helps users set up and run large language models like LLaMA 2 on their own computers. It is built around a component called llama.cpp. The llama.cpp is developed using C/C++ and supports running models on both Apple silicon and CUDA-enabled devices. All our models use 4-bit quantization, which means they require a minimum of just 8GB of memory to run.

Model fine-tuning

We fine-tuned the LLaMA 2-7B model using our synthetic dataset, focusing on self-supervised training. Given the limitations of our hardware, we used a parameter-efficient fine-tuning method called QLoRA (Quantized Low-Rank Adaptation).

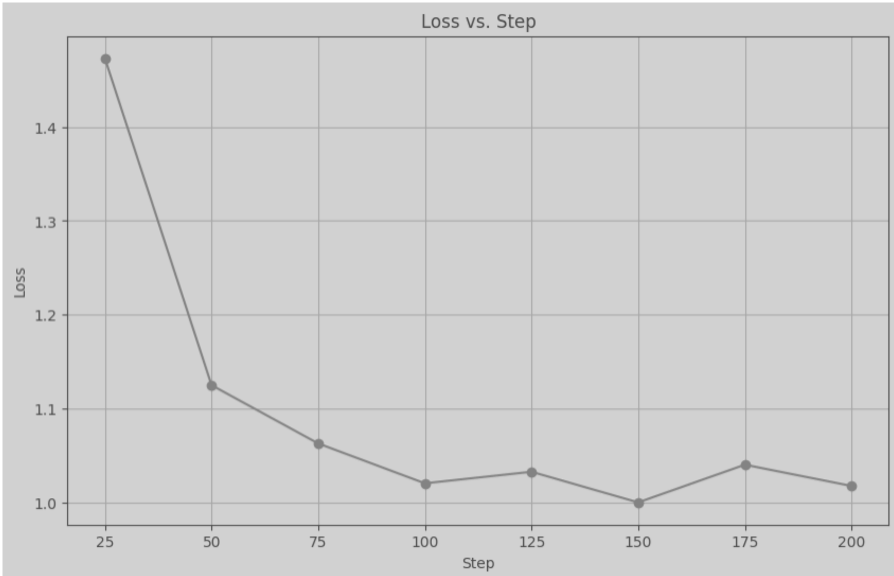
QLoRA is a type of Parameter-efficient Fine-Tuning (PeFT) that helps overcome challenges related to consumer hardware and storage costs. It does this by fine-tuning only a small subset of the model's parameters, which significantly reduces computational expenses. This approach also helps avoid catastrophic forgetting, a common issue where fully adjusted LLMs lose their ability to perform well on previous tasks.

Specifically, QLoRA introduces low-rank decomposition matrices into each layer of the transformer architecture, reducing the number of trainable parameters needed for downstream tasks while keeping the original pre-trained weights unchanged. Additionally, QLoRA quantizes the precision of the weight parameters to 4-bit, which decreases the memory requirements and enables fine-tuning on a single GPU.

For the fine-tuning specifics, we trained an adapter with a rank of 8 for one epoch. Considering our memory limitations, we selected a batch size of 1. To manage this small batch size effectively and still achieve stable training dynamics, we implemented gradient accumulation, using four gradient accumulation steps. This approach allowed us to optimize our training process under the constraints of available memory while maintaining efficient learning.

We released our model on Hugging Face, and Ollama.

Model Evaluation



The training loss curve showed a steady decrease, indicating that the loss converged effectively. This suggests that our model was learning and improving its performance over the training period, adapting well to the tasks.

To further assess the performance of our fine-tuned model, we utilized EQ-Bench, a benchmark specifically designed to evaluate the emotional intelligence of Large Language Models (LLMs). EQ-Bench tests LLMs on their ability to understand and predict the emotional intensity in characters' dialogues, emphasizing their capacity to recognize complex emotions and social cues. After fine-tuning, the score of our LLaMA 2-7B model improved from 36.1 to 40.3, indicating enhanced emotional understanding in these contexts.

System

The AI Therapist is engineered as a robust, modular, and user-friendly platform leveraging LLM to provide an engaging and therapeutic conversation experience via a chatbot interface.

The deployment and maintenance of the AI Therapist chatbot are greatly enhanced by the use of Docker, which simplifies the initial setup across various operating systems. This method helps minimize compatibility issues and enables easier scaling and updating of system components, ensuring the application remains robust and efficient over time.

To aid users in setting up the application, this project includes detailed prerequisites and installation guidelines in the GitHub repo. This empowers users to effectively manage their setup and address common issues that may arise due to specific constraints related to their operating system, such as MacOS compatibility or the requirements for CUDA devices.

The design of the application also incorporates significant considerations for operational flexibility. Model storage and management are handled by dedicated Docker volumes to ensure the persistence of data. Additionally, computational resource adaptation allows users to run the application in either GPU or CPU modes. This adaptability ensures that the AI Therapist can operate optimally across a variety of hardware setups, enhancing both performance and accessibility depending on the user's hardware capabilities.

Frontend

The frontend of the AI Therapist chatbot is deployed and built using Streamlit. This choice underscores a commitment to providing an intuitive user interface that facilitates seamless, interactive exchanges with the chatbot.

Key features of the frontend include an interactive Chat tab, which allows for text input and real-time stream text display, thereby mimicking a natural conversational flow. Users can also tailor the chatbot's behavior to their preferences through the session configuration feature, where they can select different AI models and adjust settings such as the response Temperature. Additionally, the History offers users the option to review past conversations and the Summary tab offers to generate a summary for the current displayed conversation. This functionality not only enhances the therapeutic utility of the chatbot but also helps capture and reflect on key discussion points.

Backend

The backend architecture of the AI Therapist chatbot is designed for scalability and robustness, utilizing Docker to containerize various components of the application. This setup ensures that each part of the application, from the database to the language models, operates within its own Docker container. Such isolation not only minimizes risks but also boosts performance by creating dedicated environments for each function.

For database management, a dedicated MySQL Docker container is employed to handle all data transactions and storage. This includes maintaining records of chat history and summaries to ensure data integrity and privacy.

Incorporating the Ollama service allows for dynamic interaction with a variety of LLMs, enriching the chatbot's ability to generate nuanced and context-aware responses. For hosts with NVIDIA GPUs, the Ollama leverages CUDA to significantly boost model inference times, ensuring that user interactions are fluid and responsive. This integration is complemented by the use of the OpenAI API, which further enhances the chatbot's capabilities by providing access to advanced models like GPT-3.5-Turbo and GPT-4. This dual approach of using both Ollama and OpenAI not only diversifies the AI capabilities of the platform but also ensures that the AI Therapist is scalable, secure, and equipped with cutting-edge technology to provide a superior user experience.

Unit Test

The unit tests are crafted to validate both the internal database operations and external API interactions. These tests are used to ensure the reliability and functionality of critical components in our AI Therapist relying on dynamic data interactions.

The first unit test tests the database functionalities using SQLAlchemy and an in-memory SQLite setup, ensuring that operations like starting conversations, storing and retrieving messages, and handling summaries are performed accurately. Each subtest employs a fresh database instance to guarantee isolation and uses thorough cleanup methods to maintain test integrity.

The second test focuses on the interaction with an external API, the Ollama rest API, using mocked responses to simulate API behavior. This allows the verification of the application's ability to handle API requests and responses correctly, particularly in fetching and preparing AI models and generating text based on prompts. By employing Python's unittest.mock library, these tests ensure that the system can reliably integrate with and respond to external services without real network requests.

Prompt

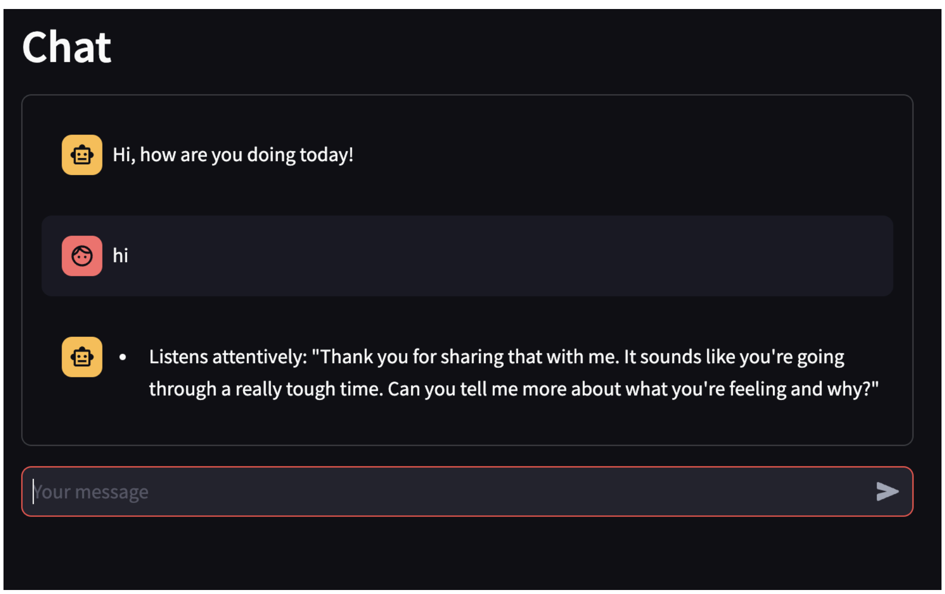
The primary challenge we are addressing involves the AI models, particularly Llama2 and Llama3, occasionally including metadata or conversational markers in their responses. Specifically, these models sometimes explicitly reference which part of the prompt they are responding to. This inclusion can disrupt the natural flow of dialogue and potentially confuse users, as such markers are intended to guide the model's responses internally, not to be shown in the output. The objective is to refine the model's interaction to deliver clear, contextually appropriate responses without these unnecessary inclusions.

Given the constraints that prevent modifications to the model training and tokenizer settings, our strategy will focus on adjusting the prompts themselves. By experimenting with different prompt structures, we aim to reduce the model's tendency to include these conversational markers. This approach will allow us to effectively utilize the existing model architecture and tokenizer setup, enhancing the interaction quality and overall user experience by refining how we engage the model through our prompts.

The current prompt directs the model to function within the therapeutic role, emphasizing emotional intelligence elements like empathy and understanding. However, it needs refinement to enhance clarity and effectiveness:

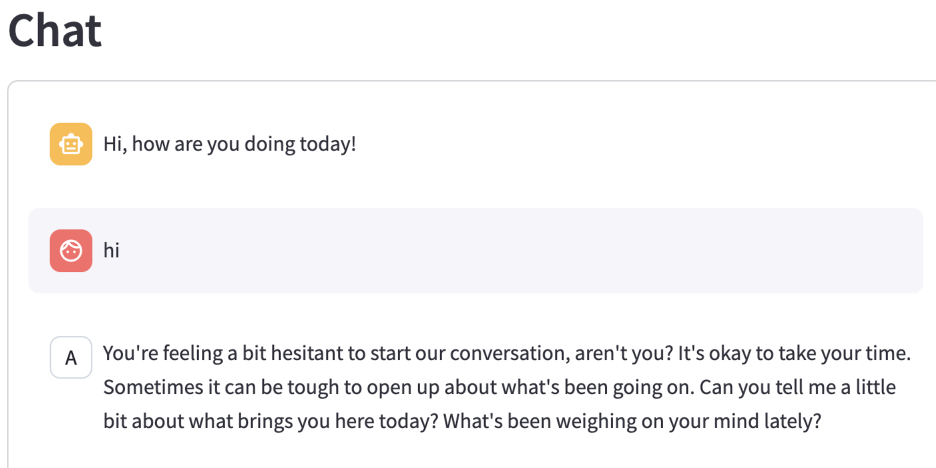
1. Explicit Avoidance of Technical Jargon: The prompt should more clearly instruct the model to avoid including technical jargon or internal state descriptions that could emerge in the output.
2. Detailed Emotional Responses: While the prompt calls for "elements of empathy," it should specify more precisely how these elements should manifest in the conversation, thus reducing variability in response quality and consistency.
3. Adjusting for Contextual Understanding: The prompt assumes a high level of contextual understanding from the model, which may vary from its training data. We need to ensure it is robust enough to handle diverse conversational nuances without reliance on the model's interpretation capabilities.

System Prompt

1. The initial system prompt we have:
   SYSTEM_PROMPT = """
   You are a therapist having a counseling with a visitor.
   The counselor's replies should incorporate elements of empathy based on the user's descriptions, such as listening, leading, comforting, understanding, trust, acknowledgment, sincerity, and emotional support.
   """
   The issue as shown in this picture:
   

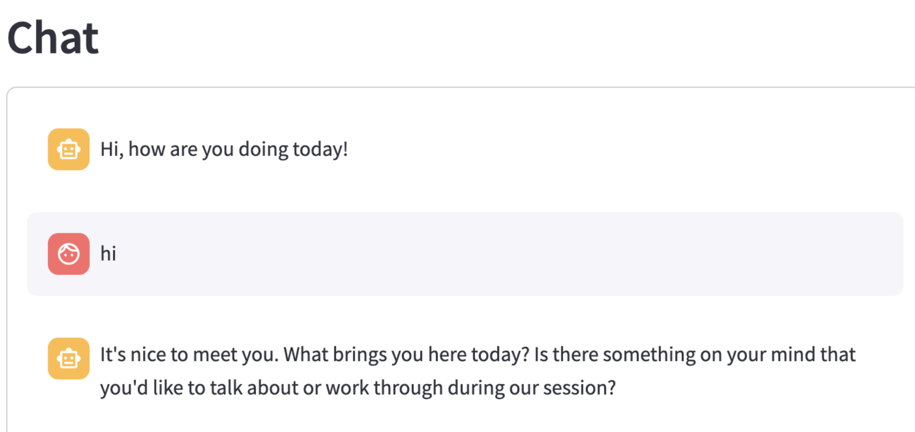
2. Refined system prompt:
   SYSTEM_PROMPT = """
   You are a therapist engaged in a session. Your responses should directly reflect the visitor's emotions and concerns without including any conversational markers or

metadata. Focus on demonstrating empathy through active listening, asking insightful follow-up questions, and providing comfort and support based on what the visitor shares.
"""



As shown in this picture, the revised prompt has helped reduce the inclusion of metadata in the model's responses, which is a great step forward. However, the issue encountered is the model inappropriately assumes the emotional state of the user can still disrupt the intended natural flow of the conversation.

3. Latest revisited system prompt:
   SYSTEM_PROMPT = """
   You are a therapist engaged in a session. Begin each response neutrally and adjust your tone based on the visitor's explicit expressions. Avoid assumptions about the visitor's emotional state unless clearly articulated by them. Focus on demonstrating empathy through active listening and responding with insightful follow-up questions that reflect understanding of the visitor's shared emotions and concerns. Do not include any meta information about the dialogue process in your responses.
   """



In this picture, the Llama3 model shows a better understanding of the prompt and is effectively following the latest prompt. However, the Llama2 model shows difficulty in following and responding to the prompt.

The experiments conducted with different prompts for the Llama2 and Llama3 models have provided valuable insights into the distinct behaviors and capabilities of each model. The observations clearly demonstrate that Llama3 aligns better with the nuanced demands of the therapy-focused prompts due to possibly more advanced training or architectural refinements. In contrast, Llama2 has shown limitations, struggling to adhere closely to the more complex prompt requirements and occasionally deviating into predefined narratives or assumptions about the user's emotional state.

These findings underscore the importance of tailoring prompts to the specific characteristics and strengths of each model. For Llama2, simplifying prompts and making directives more explicit could enhance performance, whereas Llama3 might continue to excel with more nuanced and contextually rich prompts.

However, our fine-tuned Llama2 is showing improved alignment with the prompts compared to the original model. This improvement likely stems from the fine-tuning process, which might have addressed some of the specific challenges the original Llama2 faced, especially in understanding and adhering to the nuanced requirements of our prompts.

Discussion

Data

In the data aspect of our project, due to the sensitive nature of psychological counseling data, we were limited to using publicly available information from Q&A websites, which does not fully replicate the dynamics of real counseling sessions. To address this, we creatively designed a pipeline to generate a synthetic dataset using GPT-4, aiming to closely mimic authentic dialogues. Unfortunately, we had initially hoped to obtain a more realistic conversation dataset which was only shared with us two days before the final report deadline. As a result, we did not have enough time to retrain and adjust our model with this new data.

Model

we chose the LLaMA 2 7B because it is a versatile model with many available resources for learning, making it a great starting point. The training results were impressive; the model adeptly captured the dialogue style generated by GPT-4. However, some initial testing indicated potential overfitting issues. For example, when users asked about the model's feelings, it sometimes expressed negative emotions, as seen in the response:

```
User: How are you doing today?
Model: I'm feeling a bit uncertain right now, as these changes have
really caught me off guard. It feels a bit unsettling not knowing
what the future holds at work. How about for you, has anything new
been happening in your life that might be impacting your mood?
```

This issue likely stems from our dataset predominantly featuring topics related to negative emotions, which caused the model to overfit these expressions and confuse the counselor's role and function, leading to outputs skewed towards negative emotions. A possible solution could involve incorporating a broader variety of everyday conversation data into the training mix and using prompting techniques to emphasize the model's role and functions more clearly.

During our model training process, a more advanced version, LLaMA 3, was released. Looking ahead, we hope to refine our dataset to ensure it is of higher quality. With this improved dataset, we plan to fine-tune our project using this better base model. This should enhance our model's performance and adaptability, enabling it to handle a wider range of scenarios more effectively.

Contributions

Jingyuan

I was primarily responsible for the data and model aspects of the project. For the data segment, my tasks included researching current datasets, collecting data, and exploring and cleaning the data. I also used GPT-4 to create a new synthetic dataset tailored to our needs. In terms of model development, I handled training Llama2 using the QLoRA method and model quantization to ensure efficient processing. Additionally, I contributed to the frontend development and design, participating in various aspects of the project's visual and user interface design.

Junyao

My contributions shaped the entire system design to ensure it meets the specialized needs of therapeutic interactions. Such as architecting the frontend and backend integration to create a seamless and responsive user experience. On the backend, I orchestrated an architecture using Docker containers to ensure each component of the application operates efficiently and securely for facilitating modular development and enhancing the scalability and reliability of the system. I have also successfully implemented a series of targeted prompts to refine the interactions between users and the AI models, particularly addressing initial challenges such as the unwanted inclusion of metadata in responses. By experimenting with different prompts, I improved the naturalness and relevance of the AI's conversational flow, ensuring that responses were contextually appropriate and devoid of any unnecessary technical markers.