

ECE 1786: Project Final Report

Amir Hossein Mobasheri

Fahimeh Fazel

December 2023

Word Count: 1999

Introduction

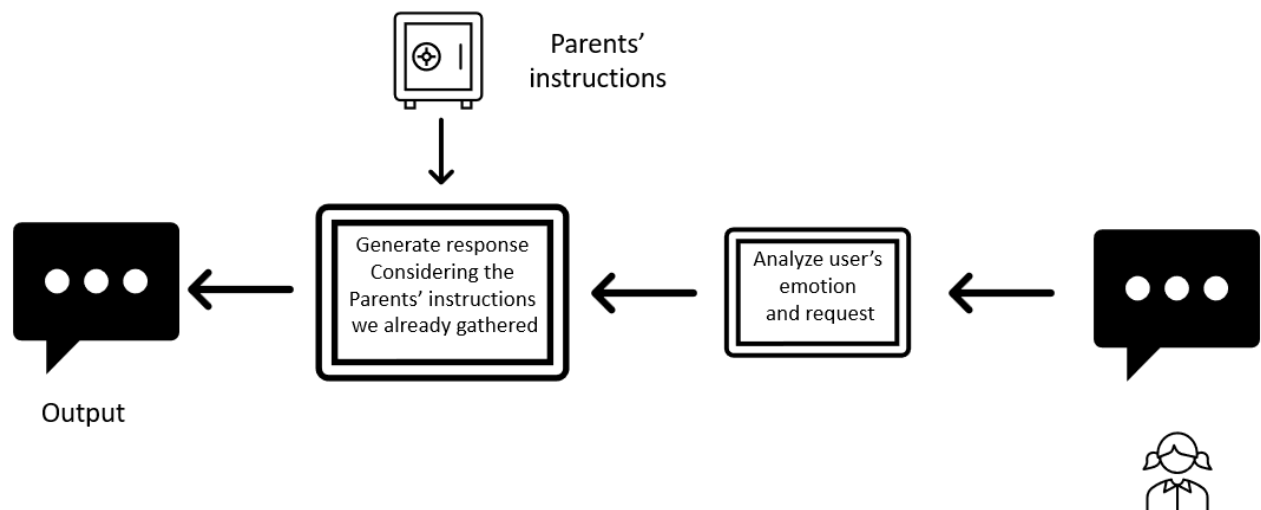
WiseOldGPT is a specialized chatbot designed for school-age children, aimed at being an engaging and supportive companion. It offers interactive storytelling, jokes, games, and riddles, while also assisting with schoolwork and providing advice on interpersonal relationships and emotional well-being. The chatbot adopts an educational approach, encouraging children to explore answers rather than providing direct solutions, reflecting its role as a wise, friendly guide.

A critical element of WiseOldGPT is the integration of parental input, allowing for customization based on each child's unique psychological profile, abilities, and needs. Parents set up the chatbot by answering questions about their child's behavior and preferences, ensuring a tailored and sensitive interaction.

The chatbot is designed to communicate effectively with children, using simple, clear language and adapting to sudden changes in conversation topics. This flexibility keeps children engaged and caters to their dynamic interests and emotional states. Additionally, the chatbot emphasizes safe and age-appropriate content, especially in games and advice, prioritizing the child's safety and well-being.

Overall model

The diagram provided illustrates the operational mechanism of our system. A more comprehensive explanation of the model's architecture is detailed in the 'Model Architecture' section of this report.



Background

The paper in [1] designed a GPT-4-based chatbot approach, prioritizing tutoring over direct solutions in "Java programming" education. This methodology extends to WiseOldGPT, focusing on nurturing problem-solving skills for children's schoolwork rather than solving math questions directly. Meanwhile, researchers [2] created medical chatbots utilizing AIML

and XML for accurate, cost-effective healthcare, inspiring WiseOldGPT's role as a supportive mentor, instead of babysitters, for alone school-age children.

Data and Data Processing

In this project, data utilization was primarily required for the development and evaluation of two classifiers, detailed further in this report. These classifiers, although based on GPT-4 prompt-engineering, required specific data for validation and effectiveness assessment.

The first classifier developed is an emotion classifier. Its purpose is to categorize user inputs into seven distinct emotional states: Angry, Disappointed, Surprised, Neutral, Sad, Excited, and Happy. To test this classifier, a total of 67 examples were generated using the OpenAI playground, with an equal distribution of ten examples for Disappointed, Surprised, Neutral, and Excited, and nine for the remaining emotions.

The second classifier focuses on command interpretation. It classifies user inputs into one of three categories: 1) Request for a story, 2) Seeking advice, and 3) Casual conversation. For generating test data, a dual-method approach was employed. Initially, 20 examples for each category were created through a system-generated prompt leveraging GPT-4. However, it was observed that the classifier sometimes confused requests for advice with casual conversation, particularly in scenarios involving complaints. To address this, an additional set of 20 examples were manually crafted for further evaluation and refinement of the classifier.

These examples, along with further details on the classifiers, are documented in the `commandClassifier` and `emotionClassifier` notebooks, which are available on GitHub. Accuracies are reported later in the report.

Model Architecture

Conversation mode:

The architecture of our conversation program is centered around a key object called 'user', which is instantiated at the start of the program. This object encapsulates crucial user-related data such as name, age, parents' instructions, chat history, and other pertinent conversation details. The program retrieves this information from a file titled 'profile.txt', focusing exclusively on user-specific content. The user's profile is then fully fleshed out by filling the object's variables and strings with the gathered data.

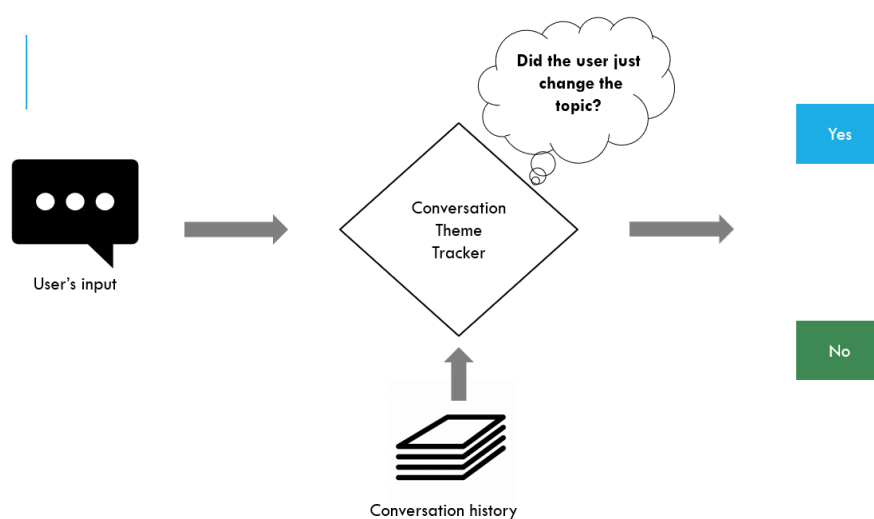
At the inception of the user profile, a default system message is constructed. This message encapsulates all the personal information of the user, the user's most recently detected emotion, and the entirety of the parents' instructions. This default message serves as a foundational element for further message processing within the system and will be attached to specialized instructions.

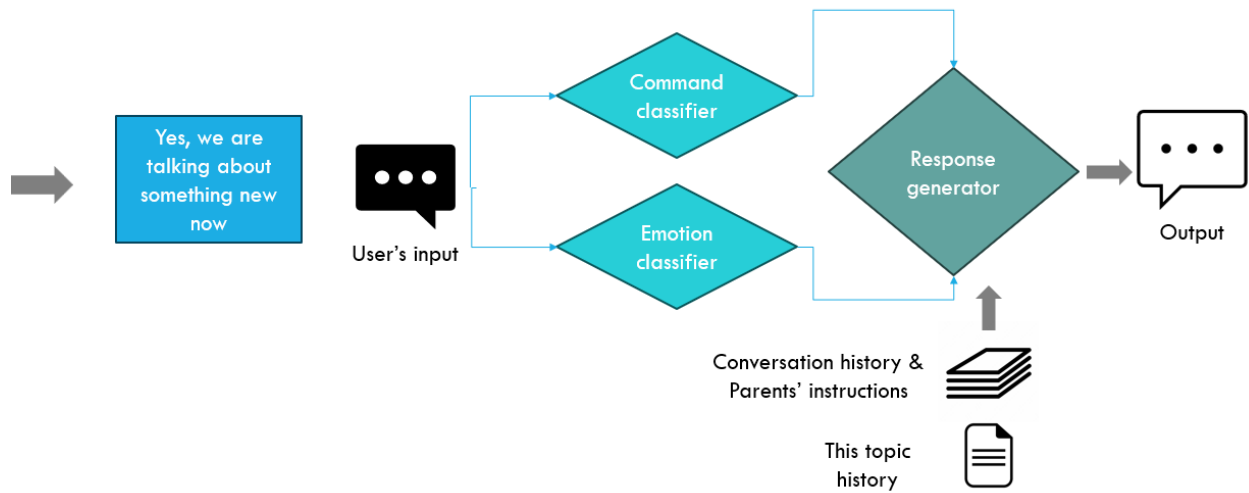
The conversation aspect of the program is administered by a function named 'manageConversation'. When a new message from the user is received, the 'phaseChanger' function is called upon to assess if the message is a continuation of the previous topic or an initiation of a new one. This function operates by analyzing the entire conversation history and the most recent interactions, utilizing properties like 'user.storyBuffer', 'user.adviceBuffer', and 'user.casualBuffer' which store messages related to the current theme. These buffers are lists of strings that each time we attach them to the system message given to GPT-4.

If a change in conversation topic is identified, 'user.conversationPhase' is set to 'idle', and the new message is forwarded to two classifiers: 'emotionClassifier' and 'commandClassifier'. The 'emotionClassifier' assigns the user's emotional state to categories such as Surprised, Disappointed, Neutral, Angry, Sad, Happy, or Excited, updating 'user.emotion' accordingly. Simultaneously, the 'commandClassifier' discerns the user's intent, be it for storytelling, advice, or casual conversation. Corresponding handler functions ('storyHandler', 'adviceHandler', or 'casualHandler') are then activated. These handlers, along with the 'clarifyHandler' function (used when intent is unclear), construct new system messages for GPT-4 by appending the default message with specific prompts tailored to their functionality. If the user's intent is clear, user.conversationPhase will be set according to the output of the commandClassifier.

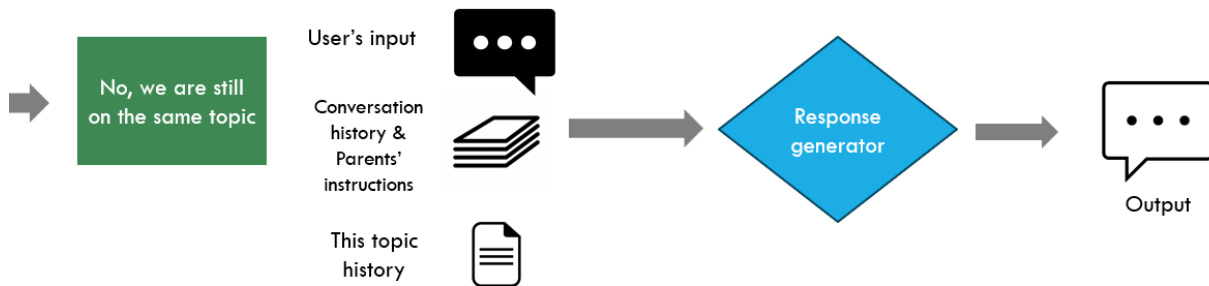
Conversely, if the user's message is a continuation of the current theme, the program bypasses the classifiers and activates the relevant 'talk' functions ('storyTalk', 'adviceTalk', or 'casualTalk'), determined by 'user.conversationPhase'. These functions distinguish themselves from the handlers by including specific system messages that provide GPT-4 with recent conversation history and context, again integrating the default message for continuity and relevance.

Importantly, each time the 'emotionClassifier' is called into action, the default message undergoes reconstruction to reflect any changes in the user's emotional state, ensuring that the system message remains current and accurately always represents the user's profile and emotional context.





Response generator is the placeholder for one of: {storyHandler, adviceHandler, casualHandler}



Response generator is the placeholder for one of: {storyTalk, adviceTalk, casualTalk}

Information Collection from Parents:

As mentioned earlier, we gather information about children from their parents. To commence this process, parents must set a password during their initial interaction. Upon entering the correct password, they can proceed to input information about their children or edit previously submitted details. Subsequently, they are prompted to provide additional details about their children. Acknowledging the potential for errors in data entry, we incorporate validation checks during each conversational prompt with parents.

For example, when inputting a name, we ensure adherence to reasonable linguistic conventions, eliminating entries like "fffa" that are impractical to pronounce. Similarly, for age, parents are restricted to entering a digit between 8 and 18, and deviations prompt a correction request. When describing personality traits, immediate emotions are discouraged; instead, parents are guided to articulate enduring characteristics like "depressed."

Upon concluding the conversation, parents have the option to review and edit their responses. The chatbot meticulously validates each answer during interactions. After completing the

editing process, the child's characteristics are compiled into a list based on parents' input. This compilation proves crucial for subsequent interactions between the chatbot and the child. This is paramount as the chatbot must adapt its behavior according to the child's specific needs.

For instance, if a child faces challenges related to autism, characterized by difficulty in self-directed play without specific instructions, the chatbot can adeptly provide comprehensive explanations. Information gathered from parents, along with other pertinent details, serves as a valuable resource for tailoring interactions with children.

Baseline model

We adapted GPT-3.5 to serve as a chatbot for school-age children, focusing on creating engaging, amusing, and supportive conversations. Recognizing that many children often face solitude due to busy parental schedules, we tailored ChatGPT's language and responses to resonate with this audience. This phase aims to create a baseline model for benchmarking against our developed chatbot's performance.

Quantitative Results

We designed five conversations, each involving interactions with both GPT-3.5 and WiseOldGPT functioning as chatbots. It is worth mentioning that we structure the prompts in a manner that allows for a direct comparison of responses. In essence, each child expressed similar feelings and commands across the conversations. Subsequently, we conducted ten conversations, encompassing five with both GPT-3.5 and WiseOldGPT. Three evaluators were engaged to assess the results based on three metrics:

User-friendliness

Effectiveness in providing helpful responses for a school-age child

Ability to resonate with the user

The evaluators were tasked with choosing the preferred option based on the above criteria, particularly considering which is more suitable for children.

Since, we had 5 conversations and 3 people, there were 15 votes to win. We won 13 votes, lost 1 and one was draw. Group one means the conversation with the baseline model and the group 2 refers to the conversations with WiseOldGPT.

The results of the voting:

Evaluator 1:

Conversation #1	Conversation #2	Conversation #3	Conversation #4	Conversation #5
WiseOldGPT	WiseOldGPT	None	WiseOldGPT	WiseOldGPT

Evaluator 2:

Conversation #1	Conversation #2	Conversation #3	Conversation #4	Conversation #5
WiseOldGPT	WiseOldGPT	WiseOldGPT	WiseOldGPT	WiseOldGPT

Evaluator 3:

Conversation #1	Conversation #2	Conversation #3	Conversation #4	Conversation #5
WiseOldGPT	WiseOldGPT	WiseOldGPT	WiseOldGPT	ChatGPT

The summary of these comments is reported in the qualitative results part

Qualitative results

This is an example of a part of conversations between chatbot and a child.

Bot:

Susan, can you guess what it is with yes-or-no questions?

Child:

is it a living thing?

Bot:

Nope, Susan, it's not a living thing. Good question though! Your turn to ask another one. Keep going!

Child:

is it white?

Bot:

Sometimes it is, sometimes it isn't. You're doing well! Keep asking those questions.

Child:

can we see it at night?

Bot:

Ah, great question, Susan. No, we generally can't see it at night. You're really getting warmer! Try another question.

The analysis of the comments of the evaluators shows a strong appreciation for its child-centric design, with strengths in providing supportive advice, empathetic responses, age-appropriate communication, and educational value, particularly in simplifying complex concepts. However, it occasionally intensifies a child's frustration and uses inappropriate wording, indicating a need for more sensitive language choice. Overall, compared to Group 1, the evaluator prefers Group 2's model for its supportive nature and attention to children's emotional and developmental needs, though it requires refinement in maintaining a consistently supportive tone and appropriate language.

For example in this snippet of conversation the model is very encouraging and positive and engages the child in the game.

Discussion and learnings

We successfully use GPT-4 for a variety of tasks: data generation, classification, response generation and etc. Therefore, we can say GPT-4 is a very powerful tool capable of doing various tasks even if they seem not doable by GPT.

We believe an integral part of a chatbot is to control the flow of conversation. phaseChanger function was a key function to have a coherent conversation. Without this part, the answers were mostly irrelevant.

Sometimes model sounds fake because it is too kind. Maybe we need to do some additional training by showing lots of conversation examples to the GPT inside the system message and force it to mimic them. This way it can sound more natural.

The supportive tone and the way that chatbot walks the user through the solution is our best property.

Individual Contributions

Fahimeh's responsibilities included coding parental instructions in the Gradio interface for the WiseOldGPT chatbot, enabling a password system for parental access, and allowing parents to input and edit information about their child's personality with GPT's assistance. She developed

the Emotion Classifier, GPT-4 prompts, and served as an instructor for ChatGPT, creating ten conversations for evaluation to improve the chatbot.

Amir Hossein focused on prompt engineering for the commandClassifier and designing the conversationManagement function. This included the phaseChanger function for detecting topic changes in conversations and developing 'handler' functions (storyHandler, casualHandler, adviceHandler, clarifyHandler) for initial response generation. He also created 'talk' functions (storyTalk, adviceTalk, casualTalk) for maintaining coherent conversation flow, essentially overseeing the functions controlling the conversation's progression.

References:

1. H. Martinus and L. Butgereit, "Reigning in GPT-4 to Tutor Java Programming Pupils and Students Instead of Just Providing Answer Algorithms – a Prototype," 2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES), Nairobi, Kenya, 2023, pp. 000165-000168, doi: 10.1109/INES59282.2023.10297873.
2. 2.P. Srivastava and N. Singh, "Automatized Medical Chatbot (Medibot)," 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2020, pp. 351-354, doi: 10.1109/PARC49193.2020.236624

Permissions:

Fahimeh:

Permission to post video: yes

Permission to post final report: yes

Permission to post source code: yes

Amir Hossein:

Permission to post video: yes

Permission to post final report: yes

Permission to post source code: no