Psychologist Chatbot

Using LLM and Revolutionary Methods to Predict

User's Big Five Personality Traits

Chung-Yeh Yang

Summery

To build a phycologist-LLM agent aiming for performing the Big Five personality traits test on patient, I tried five different methods of building it: **Chain of thought reasoning**, **CAMEL**, **Tree of thought prompting**, **AutoGPT**, and **Generative Agents**. Aside from the three models I was obliged to do (Chain of thought reasoning, CAMEL, Generative Agents), I chose Tree of thought prompting and AutoGPT since I believe their advantages can provide get me closer to the goal.

The task I set for a psychologist-LLM agent for every method is same: the agent will ask the users' background information, find out their Big Five personality traits by asking personalized questions, and provide feedback based on the result. Amongst five methods, some models can perfectly execute the task while others cannot. Chain of thought reasoning, AutoGPT and tree of thought prompting are the three most efficient methods that can accomplish our goal. Chain of thought reasoning is the easiest model to build with, and it can create a well-performing chatbot as long as a good prompt is given. AutoGPT might be the most intelligent agent amongst all of them; all we need to do is to assign a simple task and several specific goals to achieve, and AutoGPT will provide us with a chatbot specified in that task. It took some time to fine-tune the model though, since the agent would not always execute the task. Tree of thought prompting is like the advanced version of chain of thought reasoning. The main advantage of it is that it can simultaneously come up with several possibilities of next action, and we can leverage the agent to help us choose the one that has the highest predictability. It is an excellent way to increase the predictability of the chatbot. CAMEL and generative agent might not be the first choice of this project since their expertise is not user-to-AI chatbot. Despite that, I have generated conversation between two AIs using CAMEL and try to create characters using generative agent. Both methods are interesting and insightful.

The challenge that I have encountered throughout the whole project is the evaluation of the models. All models meeting the desiderata including succinct, professional, unobtrusive, and efficient dynamic questioning. Interestingly, I didn't include all desiderata in every method but all methods automatedly meet them after some tastings. I am thinking that maybe in LLM, the word "psychologist" has already come with default settings such as professionalism and unobtrusiveness. However, the tricky part is the way to evaluate the effectiveness of them. The ideal method is to have user feedback of each model, but it is difficult to realize. After some thinking, I realize to see if the method balance three properties: time to completion, adaptability and predictability may be the most realizable means for evaluation. I will elaborate my envision of the evaluation in the next part.

Overall, five models have their own advantages and room to improve. Considering the properties of each method and their performance on time to completion, cost of the model and adaptability, I can rank the five models from the highest to the lowest in terms of their character as the psychologist-LLM chatbot that performs Big Five personality traits test: tree of thought, chain of thought reasoning, AutoGPT, generative agents, and finally CAMEL. Although the power of AutoGPT as a customized task-based model is stronger compared to other methods, the process to finetune a perfect model is way too difficult compared to tree of thought prompting and chain of thought reasoning. Tree of thought prompting and chain of thought reasoning both performed well in terms of adaptability and natural flow, but the former had a higher predictability. Although part of time to completion will be sacrificed, it perfectly balances the difficulty of implement it and performance on the metrics. I will further elaborate how each method works and their evaluation in the final part of this report.

Operationalizable metrics

Coming up with operationalizable metrics is one of the thorniest challenges to address when building this psychologist-LLM chatbot. In terms of the **predictability**, unlike some studies where there's a definite answer for the fixed questions so that the accuracy of the answers can be an objective metrics to be evaluated, this research personalizes the questions based on users' background and there's no exact right answer for each. **User feedback** is the best practice we can do to determine the predictability, and if the psychologists are the users and tell if the result makes sense is better. We can find several psychologists to act as the user and interact with chatbots of different methods and rate the performance of the chatbot from 1 to 10, based on whether the process resembles that in reality and if the result is accurate. Now, without the user feedback, all we can do is to implement some actions that we believe can improve the predictability of the agent, such as tree of thought prompting. We can only dive into the properties of each method to tell if the method is likely to produce good productibility.

Another metrics that should be taken into consideration is **time to respond**. Time to respond is a metric that can determine the cost of the agent. More time an agent responds to a question, more cost would the agent take. The easiest way to calculate the time to respond is use the "**time()**" in python to see how many seconds it takes for agent to answer a certain question. We should do this before building methods into a chatbot. We first implement the method we would like to take and ask some questions that a psychologist would encounter during a session and time the seconds the agent responds for each question. We can then compare which method did a greater job. However, this means has one blind spot: the text length of the output of each model would be different, and it will impact the time to respond. Some methods might have a shorter responding time with a simple answer, while others have more detailed answers with longer responding time. We can either **limit the answer to certain words or divide the time by the tokens of answers**. In both ways the metrics would be more proper.

Adaptability is also one metric I believe is important to be taken care of. One of the main differences that distinguish this LLM chatbot and online assessment test is its ability to construct coherent conversation with user and cater to users' need. Even if the chatbot is instructed to ask a set of questions, it should have the ability to change the next step based on users' answer. One way I can think of to evaluate the adaptability of the model is to **brainstorm a set of special scenarios** that the question-asking session need to be halted or changed. The we can input these scenarios through the users' response and see how the AI will respond. These scenarios include users don't understand the question; users are not comfortable with the questions; users request to halt the session; users ask a question unrelated to the questions AI asked, etc. If AI acts naturally and resembles what a real psychologist would do, the adaptability of the method should be fine. To obtain an even higher adaptability score, the AI should be able to **ask personalized questions** that is related to the user's background instead of general questions and **ask follow-up questions** if it is not sure about users' answers.

Other factors such as question efficiency and conversation flow are metrics can be implemented as well, but I believe predictability, time to respond and adaptability are three metrics the most important. Not all of them can be simply implement or calculated, but obtaining these metrics can substantially benefit the effectiveness of the chatbot.

Chain of thought reasoning

Chain of thought reasoning is an easily implemented model to act as a psychologist agent compared to the other ones, so I chose to try this method at the beginning. The essence of chain of thought reasoning is to instruct the model to follow the steps we provide and produce the outcome we expect. In order to do that, there are three main steps to construct a chain of thought reasoning model: import OpenAI, construct a step-to-step prompt, with some examples that help the chatbot to follow, and build a chatbot, which implement the prompt as a system message.
I selected GPT 3.5 turbo as the LLM since it can act as a chatbot very well. Then we can enter the key step to construct a good chain of thought reasoning model: write a good prompt. Since it is a psychologist model focusing on the Big Five personality traits, at the beginning of the prompt, I instructed the system to act as "a phycologist that help users to know themselves better by conducting a Big Five personality traits analysis". Then I wrote some general instructions to train LLM, such as giving feedback based on the result. After that, the chain of thought reasoning was implemented. Instead of instructing the system to predict user's personality traits in one lengthy paragraph, I separated them into five steps, each of which has a clear and concise instruction, along with some examples that the system can mimic. Furthermore, since this model is required to be professional and respectful, I wrote the guidelines such as "Remember that you are a professional psychologist that should maintain a professional tone" in the instructions. Eventually, I asked the system to follow the steps I provided to build a chatbot.

The first attempt works generally well. The model mainly focuses on the Big Five personality traits by asking simple questions, but if we ask other general related to other psychological questions, it will respond you with professional answers. To showcase the influence of a prompt, I rewrote another version of prompt, which asked the agent to know one's personal information, ask questions based on that and provide feedback. The outcome was more desirable: the chatbot will provide scenarios you may face in some questions to make the user easier to answer the questions; meanwhile, the chatbot resembles a real psychologist even more.

The practices show the main advantage of Chain of Though Reasoning: all you need is a good prompt to build a decent model. At the same time, it is the disadvantage: the quality of the model heavily depends on that of prompt. If I would like to build the chatbot that really simulate the real-life consultation with a psychologist, where a psychologist may ask more detailed questions about daily life and follow-up questions, I need a lengthier and more detailed prompt than what I wrote. However, the longer the prompt get, the more susceptible the chatbot would not follow your instruction if the prompt is not written in a perfectly built structure.

The challenge of this method is its difficulty to evaluate or find a way to improve. The only thing you can work on is the prompt, but you won't know if it's getting better unless we console the real physiatrist or ask the same questions to many users and gather their feedback. It is likely that implementing more steps to clarify the instructions even more can improve the predictability, but at the same time, the time to respond/cost of the model will be sacrificed. The key to building a well-performed model with chain of thought reasoning is to find a balance between predictability and prompt lengths.

Overall, I would say chain of thought reasoning with a good prompt will have fair estimated performance power. Based on the prototype, I believe the model is using concise and limited questions to determine a person's personality trait. Nonetheless, the number of questions the model asked is not comparable to a standard examination of Big Five personality traits or a real psychologist, so we can consider it as a simple tool for those who want to have a simple overview of their personality trait.

CAMEL

CAMEL is an interesting agent to be worked with. Instead of generating a normal chatbot, I decided to take advantage of its most unique property: Role Play Framework. CAMEL enables two AIs, one user AI and one assistant AI to interact with each other. Furthermore, we only need to write three prompts: task specification, user description and assistance description to generate a chat model. For starter, I install the camel package from GitHub, and imported required code(camel.agent). Camel.agent has automatically loaded with a LLM, so I don't need to import another one. I then used a template I could find in the CAMEL GitHub to start the writing prompt. I first used the existed role in the AI society for the AI assistant, which is a phycologist, and I wrote a simple task prompt. The result was not anticipated. The AI assistant, psychologist, responded me with the instruction of how to perform a Big Five personality traits test instead of simulating a real test, which is not aligning with our goal. It might stem from the fact that the prompt was not written clear enough, because we didn't request the assistant to ask user questions. Also, the assistant is not natural enough like a real psychologist. It listed all the means for solving one questions as bullet points, which a regular psychologist would not do.

Therefore, I decided to make the prompt clearer and added some detailed role description on user and assistance. For the assistant, I wrote "psychologist who excels at conduct personality test by asking questions", and for users I wrote "Patient who is curious about his personality traits". By making it more like a real-world scenario, the user and assistance started to interact. The user generated a profession and personal life itself and assistant would analyze what the user said. It creates a simple simulation of a Big Five personality traits test. However, the psychologist agent didn't ask the question. It's the instruction part of the user tell the user what to do. To fixed that, I wrote the task prompt more detailed by mentioning "asking questions about one's personal life", and it turned out working well. The assistant started to work as a real psychologist and asking open-ended questions about the user's personal life, such as an experience of stepping out of comfort zone. Using the input users gave, the psychologist can analyze the Big Five personality traits. Finally, I tried to change the user to an artist rather than a normal patient, to see if the conversation would be different. Surprisingly, the assistant not only analyzed its personality trait, but also gave advice based on its profession. For example, when the assistant said that the user had high openness, it also said "Your natural inclination to explore unfamiliar territories and embrace new perspectives is a valuable trait that can greatly impact your artistic journey."

Although I let AI to do my part and acted as a user, the CAMEL has shown us its marvelous strengths. One of its advantages is its convenience. Compared to Chain of thought reasoning, we don't need a step-by-step prompt written. Instead, we can write three short prompts and the system will even provide a specified prompt to better fit the LLM before running the model. Also, it is easily customized. Just like the artist example, if I change the description of the user, the psychologist will provide feedback based on your profession or lifestyle. Lastly, it simulates the real-world scenario well. CAMEL acted like a real psychologist who would ask questions in depth and dive into your personal life, but at the same time stay professional.

There are disadvantages and challenges of CAMEL though. One of the main disadvantages is its infinite conversation. There're several times in the simulated conversation that a conclusion was drawn, but the assistant would ask the same question again until it reached the limit numbers of conversations I set. Another disadvantage was that it is difficult to predict when the conversation would end. Thus, sometimes the conversation has proceeded to the

maximum count that I set, but the result was not concluded. These questions should be easily solved if it's a real person to AI scenario, but it is still a challenge in AI vs AI conversation.

In terms of evaluation, different than other methods, question efficiency are the most challenges CAMEL need to overcome. Among my multiple testing, the agents never stop their conversation within the limit I set. Time completion is another problem as well. It took much longer time for CAMEL to simulate conversation than other methods. There should be a way to limit the questions CAMEL can ask and the numbers of token the assistant can reply, to improve the efficiency. Other than that, CAMEL delivers very natural conversations and will adapt the conversation based on the context, having a great execution power as a psychologist agent.

Tree of Thought Prompting

I chose the tree of thought prompting next because of its possibility to improve the accuracy of the prediction. Similar to chain of thought reasoning, we need to prepare step-to-step instructions to the agent using LLM GPT-3.5-turbo. However, the steps that tree of thought prompting takes are different. In every step, we let the system choose a question that may give us the highest accuracy.

I first tried to only write a prompt with one implementation of tree of thought. Using tree of thought, I asked the agent to come up with three different set of questions without showing them to the user. Also, I asked it to evaluate the set and list the properties including Predictability, Time to answer and such. Then the user can choose one sets to proceed based on their needs. It turned out working well for tree of thought part; the agent successfully generated three questions set and their evaluation. However, the questions are all too easy to answer, and not integrate the background information we required the patient to fill in at the beginning. Two reasons might cause that: 1. GPT 3.5 is not a comprehensive enough LLM to process the task. 2. The prompt is still not written well enough. Since we are not able to change our LLM, I rewrote the prompt too see if it works better. I wrote a more detailed prompt by requesting the agent to include different types of questions in each set and differentiating each question set. Thanks to doing that, the agent satisfied our need even more.

Despite that, this practice is not a real psychologist does. A psychologist will base on the patient's answer to the previous question and choose best question to ask nest. To achieve that, we need the agent to handle multiple trees of thought. Two factors should be taken consideration: one is the fluidity of the conversation. It will be weird if the patient doesn't answer the question well and the psychologist jump to the next topic. Another one is the predictability of the test. By asking questions with higher predictability, we can further save our time because less questions will be asked. I let the agent determine if the last question need a follow-up question first to make conversation fluid. Then it will brainstorm three possible questions to ask next, evaluate the predictability of each one and choose the one that has the highest predictability. There's just one place needed to fix: although I asked the agent only showed one question with the highest predictivity instead of all three questions, it still shows all three questions it brainstormed. Other than that, the result is great! It did especially well on tree of thought part, since the agent brainstorms many questions at the same time and choose one that user should answer.

From the three models I tried on tree of thought prompting above, we can observe two advantages of it:

1. If its evaluation on predictability is accurate, the accuracy of the result will be much improved than a regular chat model. I cannot guarantee that on the gpt-3.5 model, but I believe gpt-4 has a strong capability to do that.

2. It will save the user's time to answer the questions. Since the agent will provide questions with higher predictability, the questions asked will be substantially reduced. Users can thus answer fewer questions to finish the test to save time.

The disadvantage is evident too:

1. High reliance on prompt writing: Like chain of thought reasoning, tree of thought needs a good prompt for satisfactory results. Although I have modified and rewrote the prompts many times, the output of the chatbot is not perfectly desirable.
2. High Cost: Tree of thought will sacrifice the cost to exchange the precision of the model. For each question, the agent will go through the step of generating three questions and evaluation, which will increase the cost of the LLM-agent.

Similar to chain of thought, the challenge of tree of thought is its evaluation part. Unlike the tasks like Game of 24 which have a correct solution to reference, the psychologist-llm lacks a standard answer to reference, unless resorting to real psychologists' feedback. Nonetheless, there's controllable parts of tree of thought, which are the total number of trees and number of splits of each tree. We can balance the cost and precision through controlling them.

AutoGPT

I then proceeded to AutoGPT. The installation of AutoGPT is very different from other methods. Instead of importing it to python, I downloaded the whole GitHub repository, added my open_ai token in .env file and run the GPT with run.sh file. It seemed to be the most optimal method to implement this psychologist model since it can customize a model specified in one field. It should be easily implemented as well since all we need to do is to input the task of the agent and up to five goals of it. However, it's more complicated than it seemed. After setting up the whole model, I first ran a simple model with only the description "act as a psychologist and finds a user's Big Five personality traits act as a psychologist and finds a user's Big Five personality traits". The result is not as expected. The agent was stuck during the part of generating questions. It kept using Chrome to search for online Big Five personality traits test and the validity of different tests. Even I authorized the command to search online, I didn't get any valid result. Although the action is understandable, it should not be what a psychologist acts. Rather than searching online, I was hoping autoGPT can take advantage of GPT to brainstorm questions. Also, I expect the agent asked the users' background first, and then asked personalized questions based on their background to find out their personality traits. I tried to change the AutoGPT into manual mode and insert some goals such as brainstorm questions based on users' background using LLM, but the result was similar. It started using Chrome to search for online test again. Should I find a way to disable autoGPT to search online?

To see whether doing that can generate better contents, I disabled the web_search command by editing the .env file. I first only input a simple description of the GPT without any goals. The result was not bad. The agent didn't ask the specific information about patient or ask questions. Instead, it asked really simple questions, such as "rate your openness from the scale 1-5" to determine each personality trait.

The model was far from ideal though. I want the model to achieve several goals: first, it should simulate a real counseling session, so it should not cut to the personality test at the beginning; instead, ask for patients' background information and confirm they're here for finding out their traits. Second, the questions were too conceptional to answer. We should request the system to ask questions that is more applicable in real world scenario; lastly, we want the conversation between AI and user be written in a text file, so that it was more readable. We could

do that by either expanding our description of the GPT or manually setting five goals needing to achieve. After numerous times of attempts, I resorted to the first method because the AutoGPT will help you generate goals based on your description, and these goals were always better than I could write. The final version was much better than the first one: it would ask my personal information first, and started asking questions that identify my Big Five personality traits. Although the questions were still conceptional, when I asked if the agent could apply the questions to a real-world scenario, it did for me. Also, it provided me with very constructive and insightful feedback. It didn't automatically print the conversation as a text file, but when I asked it, it created a text file that store the result, feedback, and advice. Although it is not exactly as expected, it did a great job as a psychologist-chatbot.

The stability of AutoGPT as an implementation of a chatbot is the most disadvantage it faces. It is understandable though, provided that we use fewer words to generate the agent with same goals compared to other methods. Also, chatbot is not the main usage of AutoGPT. Its power aims to complete and solve the given complicated tasks. However, if we successfully generate a chatbot, the adaptability and predictability of AutoGPT are better than a simply built chain of thought reasoning agent. The tone of AutoGPT is natural, and it would generate a professional response if I said something unrelated in the middle of conversation. More importantly, its strong power enables it to act as more than a simple chatbot. For example, we can output the whole conversation between the chatbot and users into a text file, which can record every visit of a patient; it can also create a pandas data frame to store the user's testing result for record. If we want to analyze a celebrity's personality trait, we can take advantage of the google_api to extract the celebrity's information online and simulate a counseling session.

Generative Agents

Generative agents are a revolutionary method that enable different AI characters to interact with each other in a created world. They even have their routinely schedule, which is suspect to change if they bump into other AI characters. I didn't have any idea how I can implement this method into this research at the first glimpse of the paper. Then I think of one application that generative agent can do but others cannot: psychologists in the real world can create a virtual psychologist online themselves.

Say there's a counseling psychologist named John owns a clinic in a small town in Massachusetts. He has wife and two kids, and he likes to play tennis with his friends during weekend. He's well known in the town he lives because he's the one and only psychologist there. He excels in analyzing patients' personality trait and give advice. Since he works in regular hour and seldom need to work after time, his weekly schedule is usually fixed. Using the information given above, we can create an agent whose name is John and write a prompt. We can also write a schedule of him, but we will skip that part for now. Instead, we assume he's always in clinic and treating patients.

We have another character named Claire, who is an elementary school teacher. She did have a passion for teaching but teaching the same material repeatedly for 7 years in a row made her monotonous. She asked the supervisors if she could change the grade to teach, they declined with the reason "there's no capacity where you can fit in right now". Even worse, during your recent annual evaluation, you got the worst score you had in 7 years. You were really, really frustrated and doubted. She decided to see Dr. John and find out what's the solution of her problem. These information enables you to create another agent.

From now, the generative agents work a little like CAMEL, since two AIs are going to interact with each other, but the difference is they are the character with given personalities. Ideally, I can assign a task to John that he can "counseling Claire, predict her Big Five personality traits, and give advice on her problems based on the result", and Clair will be given a task to "seeking counsel to John. Answer his questions as much as you can". By doing that, two AIs can interact with each other. Even more, if their weekly schedule is added, the conversation can be more natural. Say they have an appointment on Friday evening. Claire just ended a hard week in school. When John asked Claire about her recent life, Claire can talk about her hard week! This might be a breakthrough that other methods cannot achieve. Psychologists or researchers in related field can simulate different scenarios using generative agents, and thus find an optimal counseling way for each problem.

Unfortunately, I found mere sources on generative agents online, especially the python code they use. There are two python files in their original GitHub, but it didn't specify how the interaction between two AIs works. Since I had no idea how to make it happen, I decided to only create the character of Dr. John, and I will be the patient that consult to him. While this practice is way far from how generative agents actually work, it would give me an insight how a customized psychologist will interact with users. I wrote a prompt that describe him and built a chat bot. I pretend myself to be Claire and ask how should solve my problem. Everything went well. I wrote a standard procedure Dr. John would follow during a counseling session, and the whole chat followed that process. My description about John's family and hobbies were not mentioned though. It is understandable because psychologist would not share their personal information to patients. In terms of the metrics, the reaction time and adaptability of this customized psychologist seems similar to those of chain of thought reasoning.

After the testing, I firmly believe the concept of generative agent can be implemented in this research. It is very flexible and easily customized. Imagine there's a psychologist who just opened a clinic. People who haven't visited him might be afraid to make an appointment since they don't know him. What if I tell them there's a chat model online that is exactly like him, and you can try for free? People will go to the website and see if his counseling style fit them. Integrating with voice identifying system, people can even have a virtual call with him. Nonetheless, this process can be completed easily even without the utilization of generative agents. The real use of generative agents may lie in the academic research, just as I mention above, rather than a psychologist-chatbot.

In summary, the implementation of whole generative agents cannot help building a chatbot. However, utilizing the concept, we can build a customized agent that has the personality of real psychologists, with great execution power. I'm convinced it can be easily applied to many real-world scenarios and makes our life easier.