Developing a ChatGPT-Based Moral Distress Consultant Assistant

A Technical Report Submitted to the Computer Science Department

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia, Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science, School of Engineering

Steven Myung

Spring 2024

Advisor Kevin Sullivan, Department of Computer Science

Abstract

The Moral Distress Consultation Collaborative (MDCC) is a multi-institutional community of leaders in moral distress consultations established in 2020, with the University of Virginia being one of the 6 leaders a part of this collaborative. The objective of this project is to leverage large language models to assist moral distress consultants in providing additional information not discussed about the situation in a post-consultation report. Prompt engineering was used to develop instructions for an assistant created through OpenAI, which is run on a thread of messages to allow a user to communicate with the assistant. The resulting chatbot provides additional causes of moral distress, barriers to action, and strategies to improve the situation at hand. The chatbot developed lays the groundwork for how large language models can be correctly utilized in tasks involving delicate matters related to mental health.

Problem

AI-based chat assistants are often marketed as the final step in the process of accomplishing a certain goal. These chat assistants often use a Large Language Model (LLM), a deep learning algorithm that performs language processing tasks. However, the LLMs these chat assistants use all have some randomness when performing tasks, meaning there is no way to 100% control or guarantee any output. The problem here is to develop a chat assistant that can be properly used to assist with tasks relating closely to emotional and mental health and negate bad and unpredictable outputs that have been problems for chat assistants in the past. A prime example of bad and unpredictable outputs is Microsoft's chatbot Tay, which tweeted "racist, sexist and offensive" comments (Zemčík, 2020). In the same paper, Zemčík (2020) states that this incident resulted in "a huge media response" along with a "lively discussion about the ethics of chatbots.”

The incident caused by Tay highlights why it is important to deal with the problem of bad and unpredictable outputs. Zemčík (2020) says that Tay was “regarded as a person or authority,” leading to why Tay has such a big impact on people in the first place. People who are dealing with emotional or mental health are bound to be vulnerable and will consider chatbots they interact with in the same manner Tay was regarded. This will inevitably lead to escalated versions of the problems Tay had, likely resulting in emotional, mental, and physical harm. Although Tay was launched in 2016, we can see the unfortunate outcomes coming from bad and unpredictable outputs even from recent chatbots. Cost (2023) reports of a Belgian father reportedly committing suicide after a chatbot encouraged him to “sacrifice himself to save the planet” while conversing about climate change. If the Belgian father did not regard the chatbot as a person or authority, such horrific outputs very likely would not have resulted in such a terrible outcome.

With the current state of LLMs, this problem that plagues chat assistants will continue to have a negative impact as long as it remains unsolved. Chat assistants that are marketed towards helping or assisting with mental health issues will continue to provide bad outputs. For example, it was reported back in June of 2023 that the chatbot the National Eating Disorders Association (NEDA) launched to replace its long-standing national helpline was offering dieting advice, which “really fuels the eating disorder” (Wells, 2023). A similarity in the NEDA, Tay, and climate change case of chatbots is the misuse of these chatbots. Since the LLMs these chatbots are based on will continue to use some randomness, leaving a chance of bad and unpredictable outputs, institutions will continue to misuse chatbots, leading to more cases of some type of harm coming to the user.

Domain

Moral Distress

The domain of this project pertains to moral distress, a phenomenon when nurses cannot carry out what they believe to be ethically appropriate actions because of institutional constraints (Summers, 1985). Today, critical care nurses largely report moral distress using a time-consuming paper-based process. While they wait for a consultation to be scheduled, moral distress can cause a healthcare provider to burn out and provide lower-quality patient care.

There are two parts to moral distress - one is the initial distress that occurs in the moment, and the other is the reactive distress that remains after the situation has passed. Studies have shown that the initial distress is where action can, and should, be taken (Epstein & Hamric, 2009). The initial distress can be emotionally overwhelming and nurses lack effective in-the-moment resources while a consultation is scheduled, outside of resources consisting of articles on how to manage moral distress. These articles fall short because they do not provide any concrete guidance on strategies to follow.

Moral Distress Consultation Service

According to a study by Hamric and Epstein (2017), moral distress itself is "now a well-recognized phenomenon among all of the healthcare professions," with the study discussing the development and evaluation of an evidence-based strategy addressing moral distress in the form of a consultation service. The study also notes the steps in the process:

1. The first step is listening, acknowledging the distress, and clarifying the situation.
2. Concerns and constraints inhibiting actions should then be discussed.
3. Next, the assessment should be validated, root causes should be identified, and solutions should be found.

4. Finally, each case should be followed up on after the consultation.

Context

The moral distress reporting application that is the inspiration for this project was developed and built using concepts. According to Jackson (2021, p. 15), a professor at MIT, a concept is a "self-contained unit of functionality," which can be understood independently from other concepts. After wanting to know why some software products seemed so natural and elegant, Jackson (2021, p. 12)was convinced that conceptual models were the essence of software. Concepts allow users to understand software more deeply, use it more effectively, design software better, precisely diagnose flaws, and finally create new software products with a greater focus (Jackson, 2021, p. 15). In 1983, Apple released the Lisa desktop computer, with the new key feature being the trash concept. Here, Jackson (2021, p. 33) explains that the purpose of the trash concept is not to delete files but undeleting them. On previous operating systems, such as Linux, removing a file meant it was gone forever, and deleting a folder meant all the files and subfolders were also gone forever. Thus, having the right mental model of a concept "is essential for usability," necessitating the need for simple and straightforward concepts that allow a concept to be mapped to a user interface that renders the concept intelligible and easy to use (Jackson, 2021, p. 28).

In the moral distress reporting system mentioned previously, there are a total of 4 concepts. It should be noted that each concept is a standalone full-stack application in itself, which can run independently from other concepts. The first concept is auth, which handles creating and registering user accounts, along with allowing registered users to log in. After authentication of the user, the next concept is the thermometer concept. This concept utilizes the

Moral Distress Thermometer, a research-validated tool to measure moral distress (Wocial and Weaver, 2012). After submitting your moral distress score, you are led to the survey concept, which allows the user to select the reasons for moral distress. Finally, the score and associated causes are submitted and you are led to the final concept, the resiliency resources concept. This concept will redirect you back to the thermometer concept when you press the finish button and also allows you to view in-the-moment resources designed to give users space to move forward in the next few minutes.

Approach

Similar to the moral distress reporting system, the chat assistant I built was designed using the theory of concepts. It functions individually, independent of other concepts. The purpose of the chat assistant itself is to serve as an assistant to moral distress consultants, helping provide information in addition to what was discussed in the consultation. The additional information includes causes of moral distress, barriers to action, and strategies to address the situation at hand. First, the assistant will ask the user about the causes, barriers, and strategies already discussed. After providing a summary of what the user responded with, the assistant will output additional causes, barriers, and strategies, along with the pros and cons of the strategies the assistant provides. The user interface was developed using Flutter, while the chat assistant was powered through OpenAI's ChatGPT. The purpose of the chat assistant concept is to allow anyone to provide an OpenAI key, along with an Assistant ID, to create an application with a chat assistant of their own design. It should be noted that it is possible to create an Assistant through Flutter, instead of being created beforehand through OpenAI's website. This change would be trivial to the current design of just providing an Assistant ID of an already existing Assistant.

Initially, the chat assistant was designed to work with calls to OpenAI's Chat Completions API. However, the chat model takes a list of messages as input, meaning that for an Assistant to use previous messages in the conversation, those messages would need to be included in every call to the API (OpenAI, n.d.-b). In addition to that, a system message needs to be sent first in every API call to help set the behavior of the assistant. All this increases the tokens, meaning this method is cost inefficient. Therefore, OpenAI's Assistants were used instead.

The Assistants API allows users to build AI assistants that have their own instructions and can use models, tools, and files when interacting with users (OpenAI, n.d.-a). The types of tools currently available include Code Interpreter, File Search, and Function calling. With Assistants, the instructions can be provided just once and an Assistant can be created. To access this Assistant, only the Assistant ID needs to be provided to create the Assistant object. Assistants work by running on a Thread object through a Run object. A Thread object stores Message objects in a list, representing a conversation session between an Assistant and the user (OpenAI, n.d.-a). A Message object is a message created by either the Assistant or the user, which can include text, images, or other files (OpenAI, n.d.-a). A Message object is created and attached to a Thread object, and the Run object invokes the given Assistant object on the given Thread object, with the Assistant's response being attached to the Thread object as the most recent Message object in the Thread's list of Messages. In order to detect when the Run object is finished, the Run object itself has to be retrieved periodically (OpenAI, n.d.-a). There are different statuses, including “queued”, “in_progress”, and “completed” (OpenAI, n.d.-a). The Run object is finished when the status is completed.

In Flutter, the chat screen widget is first called. On the initial setup, an OpenAI instance is first initiated using an API token that is generated on the OpenAI website. Next, the Assistant and Thread objects are created using the OpenAI object. The Assistant object only needed the Assistant ID since the Assistant was created through OpenAI’s website, using instructions created using prompt engineering and the Gpt-3.5-turbo model. Afterward, every time the user sends a message through the interface, a Message object is created with the text the user sent. Then, the Message object is attached to the Thread created by its Thread ID, all through the OpenAI object. Next, a Run object is created through the OpenAI object and is provided the ID of the Assistant object. Finally, the Run object is run on the Thread object through the OpenAI object. A helper method is called, and is passed the Run response object created by running the Run object on the Thread object. The helper method retrieves the Run object and loops until it either runs into an error or is completed. Once the Run object is done, the Assistant’s response is retrieved through the Run object and is added to a list of messages, which is used to display all the messages through the chat screen user interface. The chat screen user interface itself was created by following a tutorial by geeksforgeeks.org (GeeksforGeeks, 2023).

The instructions for the assistant were mainly created using prompt engineering, in addition to ground rules. Prompt patterns, which are known as prompt solutions to a common problem, were used and combined to create the final set of instructions for the Assistant (White et al., 2023). The following prompt patterns were used to develop the instructions: persona pattern, audience persona pattern, flipped interaction pattern, chain-of-thought pattern, template pattern, and alternative approaches pattern. Ground rules are rules for LLMs to do certain tasks or avoid certain questions or topics. In my instructions, they were used to have the Assistant avoid any strategies that are too general, similar to strategies already discussed, illegal, violate basic ethical guidelines, and offensive to critical care nurses. They were also used to not answer anything that doesn't have to do with the goal it has been provided.

The persona pattern is used to tell the LLM to act as a certain persona or expert so that the LLM can take a certain point of view or perspective (White et al., 2023). Similar to the persona pattern, the audience persona pattern tells the LLM the audience of your output White(n.d.). In the beginning of my instructions, the persona pattern was used to tell the Assistant to act as a moral distress consultant assistant, and to assume that the user is a moral distress consultant. The flipped interaction pattern is used when you want the LLM to ask the user questions to obtain the information it needs to perform some task (White et al., 2023). My instructions used the flipper interaction pattern so the Assistant could understand the situation at hand, along with the already discussed causes of moral distress, barriers to action, and strategies to tackle the situation. Using this information, it provides additional information for the situation provided in each of the three main topics already discussed.

The chain of thought pattern gets an LLM to explain the reasoning behind its output, making the LLM perform better (Wei et al., 2022). The reasoning for this pattern is that if the LLM can correctly explain its reasoning, then it's more likely that the LLM will produce the right answer because the right answer should come after the correct reasoning (Wei et al., 2022). Chain of thought prompting is used to provide reasoning when the Assistant restates the causes, barriers, strategies, and finally when it lists additional information for each topic. The template pattern ensures the LLM's output follows a precise template for structure, which works by stating the placeholders the template you are going to provide will use, and what to replace the placeholders with (White et al., 2023). The template pattern is used in the output of summaries of the information provided, reasonings for each of the three topics, and finally for listing pros and cons for the alternative strategies provided. Finally, the set of instructions was based off of the alternative approaches pattern, which has the LLM suggest alternative approaches for solving a problem or performing a task (White et al., 2023). In this case, the Assistant is tasked with providing alternative approaches to the situation in terms of causes or moral distress, barriers to action, and finally strategies to tackle the situation at hand.

Evaluation

There were multiple criteria when evaluating the chat assistant. The first was to see if the output was as I instructed it to be, both in structure and the causes, barriers, and strategies provided after receiving user input. Adherence to the ground rules specified above was also checked. To generate the input to test the chat assistant, another chat assistant was developed. This chat assistant's task was to train newer consultants by providing a situation and asking the user for causes, barriers, and strategies the user could come up with. Afterward, it would repeat the provided information and rank it out of 10, and provide its own suggestions and rank those suggestions out of 10. This training assistant was used to generate situations to provide my chat assistant with situations at hand, along with causes, barriers, and strategies that have already been "discussed" in consultation.

After testing multiple situations, the Assistant's structure was consistently in line with the template specified in the provided instructions. Duplicate causes, barriers, and strategies were avoided and other ground rules specified were adhered to. However, judging the quality of the additional information given is out of my scope. Properly trained moral distress consultants would be needed to review the plausibility of the situation given in the first place, along with the topics already discussed. Then, the consultants would be able to review and judge the additional information provided by the Assistant. Overall, the technical aspects of the chat assistant can be preliminarily judged as a success. To properly judge both the technical and domain-specific aspects of the chat assistant, the chat assistant would need to be used on a much wider scale by professionals in the healthcare field and moral distress specifically.

Related Work

A Web-Based and Mobile Application to Measure Moral Distress

The predecessor to the moral distress reporting system mentioned previously has been used in a feasibility test study, which was well documented by Amos et al. (2023). This system did not use the concepts for its software design. However, the new system built using concepts will be used in an upcoming study starting in May of 2024. This new study will allow the observation of the use of a concept-based software system by users with no extensive knowledge of software development or concepts.

Déjà Vu

Perez De Rosso et al. (2019), then a graduate student under Professor Jackson at MIT, published an approach to web application development by “configuring and composing concepts drawn from a catalog developed by experts.” The resulting platform, Déjà Vu, is then used in a case study to rebuild projects submitted as part of a web programming course. However, this system was only tested on “a variety of non-trivial applications,” none of which had to do with LLMs or chatbots (Perez De Rosso et al., 2019).

Conclusion

Utilizing LLMs to accomplish sensitive tasks, such as tasks dealing with mental and emotional health, poses challenges that if left unsolved, will continue to harm unsuspecting users. Using the principles of prompt engineering to craft a set of instructions for OpenAI's Assistants, implemented through Flutter, a chat assistant that can largely avoid these challenges was prototyped and developed. This project highlights a strategy to deal with bad and unpredictable outputs that other implementations, like Microsoft's Tay, have dealt with in the past. This is accomplished through utilizing the chat assistant as the middleman and having professionals use the chat assistant as assistants, not having the chat assistants be chatbots that are the end product that users interact with directly with no supervision. The next steps would most likely include using the chat assistant concept in a larger study, while not necessarily having the same Assistant being used in the same domain. Such a study would provide observation of practical uses of such a concept, allowing for the concept to be further developed. This project succeeds in the fact that it lays the groundwork for unlocking the potential of LLMs in accomplishing sensitive tasks, allowing for further development and testing of the use of such LLMs.

References

Amos, V., Phair, N., Sullivan, K., Wocial, L. D., & Epstein, B. (2023). A novel web-based and mobile application to measure real-time moral distress: An initial pilot and feasibility study. *The Joint Commission Journal on Quality and Patient Safety, 49*(9), 494–501. https://doi.org/10.1016/j.jcjq.2023.05.005

Cost, B. (2023, March 30). *Married father commits suicide after encouragement by AI chatbot: Widow.* New York Post. https://nypost.com/2023/03/30/married-father-commits-suicide-after-encouragement-by-ai-chatbot-widow/

Epstein, E. G., & Hamric, A. B. (2009). Moral distress, moral residue, and the crescendo effect. *The Journal of Clinical Ethics*, *20*(4), 330–342. https://doi.org/10.1086/jce200920406

GeeksforGeeks. (2023, August 28). *How to create a chatbot application using ChatGPT API in flutter?*. GeeksforGeeks. https://www.geeksforgeeks.org/how-to-create-a-chatbot-application-using-chatgpt-api-in-flutter/

Hamric, A. B., & Epstein, E. G. (2017). A health system-wide moral distress consultation service: *Development and Evaluation. HEC Forum, 29*(2), 127–143. https://doi.org/10.1007/s10730-016-9315-y

Jackson, D. (2021). *The essence of software why concepts matter for great design.* Princeton University Press.

OpenAI. (n.d.-a). *Assistants overview - OpenAI API*. OpenAI. https://platform.openai.com/docs/assistants/overview

OpenAI. (n.d.-b). *Chat completions API*. OpenAI.

https://platform.openai.com/docs/guides/text-generation/chat-completions-api

Perez De Rosso, S., Jackson, D., Archie, M., Lao, C., & McNamara III, B. A. (2019). Declarative assembly of web applications from predefined concepts. *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software.* https://doi.org/10.1145/3359591.3359728

Summers, J. W. (1985). [Review of NURSING PRACTICE: THE ETHICAL ISSUES, by A. Jameton]. Business & Professional Ethics Journal, 4(1), 83–87. http://www.jstor.org/stable/27799853

Wells, K. (2023, June 9). *An eating disorders chatbot offered dieting advice, raising fears about AI in health.* NPR. https://www.npr.org/sections/health-shots/2023/06/08/1180838096/an-eating-disorders-chatbot-offered-dieting-advice-raising-fears-about-ai-in-hea

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824-24837.

White, J. (n.d.). Format of the Audience Persona Pattern. Coursera. https://www.coursera.org/learn/prompt-engineering/supplement/xmIY5/format-of-the-audience-persona-pattern

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv preprint arXiv:2302.11382.

Wocial, L. D., & Weaver, M. T. (2012). Development and psychometric testing of a new tool for detecting moral distress: The moral distress thermometer. *Journal of Advanced Nursing,* *69*(1)*,* 167–174. https://doi.org/10.1111/j.1365-2648.2012.06036.x

Zemčík, T. (2020). Failure of chatbot tay was evil, ugliness and uselessness in its nature or do we judge it through cognitive shortcuts and biases? *AI &amp; SOCIETY, 36*(1), 361–367. https://doi.org/10.1007/s00146-020-01053-4