Social Robot for the Depressed and Lonely

Based on Emotional analyzing and LLM

Charles Li
Computer Information Technology
Purdue University
West Lafayette IN USA
li4560@purdue.edu

Taehyeon Kim
Computer Information Technology
Purdue University
West Lafayette IN USA
kim4435@purdue.edu

ABSTRACT

Depression is one of the big issues in modern society. There are two main characteristics of depression that we noticed. 1) Depression is not a problem limited to a specific age group. 2) Very few people are properly treated for depression. Therefore, we would like to propose a social robot that treats depression, which is accessible to people of various ages at a low cost. The task of the proposed robot is largely divided into two parts. The first is Emotional Analyzing and the second is the Interaction using LLM part. In the first part, emotional analysis is performed using the text generated by the user, the user's facial expressions, and speech tone, and the results are integrated using LLM to analyze the user's emotions. In the second part, LLM ensures appropriate interaction with the user based on the results of the user's emotional analysis performed in the first part. This interaction consists of dialogue output and activity suggestions appropriate to the user's current emotional state.

KEYWORDS

Social robot, Depression, Emotional analyzing, Multi-modal learning, Large Language Model, Prompt engineering

1 Introduction

In modern society, physical injuries have decreased significantly due to technological advances, but on the contrary, mental illness is increasing against material abundance. Mental illness has increased as living, working environments, and human relationships have diversified, but access to depression treatment remains difficult due to a lack of awareness of mental health and difficulty in finding the cause of the outbreak. According to a 2023 report from MHA (Mental Health America) [1], in 2019-2020, 20.78% of adults were experiencing a mental illness. That is equivalent to over 50 million Americans. Also, over 10% of youth in the U.S. are experiencing depression that is severely impairing their ability to function at school or work, at home, with family, or in their social life. The most serious thing is that 54.7% of adults and 59.8% of youth with major depression do not receive any mental health treatment. Depression problem is not just a problem for a specific age group, and access to treatment needs to be increased.

Therefore, we propose the social robot to improve depression, which can be used by people of various ages with low cost. Our social robot will be implemented in a software robot (such as an AI assistant) rather than having a hardware robot form, such as a humanoid or pet robot. There are two main reasons why we propose robot in this form of software. First, our social robot's goal is to reduce the cost of production as much as possible because it aims to be accessible to a wide range of people at a low cost. If a robot has a physical form, it is directly linked to an increase in cost. If a physical form is needed for this system in the future, the system is basically composed of Python scripts, so it can be applied to various devices such as personal computers, smartphones, and robots equipped with Raspberry Pi. The second reason is that a large number of depression-treatment social robots mainly focus on the external part of the robot, which reveals limitations in intangible interactions such as conversations with users, and our project is to focus on making the robot have a more natural conversation with users.

The structure of proposed social robot is largely divided into two parts. The first part is emotional analyzing that receives various data from users and analyzes emotions. Since accurate emotion analysis is difficult with just one type of user data, multi-modal based emotion analysis is performed through various data such as voice tone and facial expressions. The emotional analysis results performed separately on each data are combined into one through Large Language Model (LLM). Through LLM, it is possible to combine emotion analysis results of different types more easily. And based on the output of emotional analyzing, our social robot proposes actions and interacts with user by natural conversation using the LLM to improve user's mental health. LLM have the advantage of being able to make conversations with users more "naturally" and "like humans" through natural language processing, but also LLM have the disadvantage that the output form is not constant. This is a major obstacle to the use of LLM in specific system structures. To overcome this, we learn LLM through prompt engineering to generate the appropriate form of output for the social robot system structure we propose.

In summary, we propose a social AI robot that analyzes the user's current emotional state and takes appropriate interactions based on it. Emotional states are assessed in a variety of ways and the results are integrated by LLM. And based on the analyzed result, the LLM learned by prompt engineering ensures appropriate interaction with users. This interaction includes appropriate

dialogue and activity suggestions based on the user's emotional state.

2 Related Works

There are a significant number of social robots for treating depression [2] [3]. But many of them are limited in that they are for a certain age group, especially the elderly [4] [5] [6]. In addition, most of research and products are mainly focusing on accessible interfaces or cute and relieved appearances. This may be valid for a specific age group targeted for research or products but is not suitable for use by various age groups.

In addition, in the point of view of emotional analyzing, one of the approaches of our project, there is pepper [7] that conduct emotional analyzing, but it mainly focuses on external parts such as natural conversation tones or natural movement. Therefore, even if they analyze users' emotions, there are limitations to interact with users because that functions are very simple and fixed in those robots.

3 Methodology

Our proposed social robot receives various data from users like voice tone, facial expressions, text messages and then performs a multi-modal emotional analyzing based on it and outputs the results. Subsequently, this output is passed on to the LLM, generate the output of conversations and activities proposals appropriate for user's particular mental state. Through this process, our social robot can detect the user's mental state and relieve the user's depression.

For this purpose, the proposed social robot has a system structure that is largely divided into two parts, as depicted in Figure 1. The first is **Emotional analyzing** and the second part is **Interaction using LLM**.

3.1 Emotional analyzing

In the context of emotional analysis, to address the unique needs of the target user group and enhance system accessibility, the system has designed the emotional analysis components as multiple independent plug-and-play modules, deviating from the direct use of a multimodal fusion approach. These modules allow the free combination of inputs from different modalities with the analysis module, catering to the requirements of individuals in the depressed and lonely population who may lack expressive capabilities in specific modalities. These particular users may be reluctant or unable to utilize a single modality such as text, speech, or facial expressions due to functional barriers. For instance, individuals with depression may not be inclined to convey emotions through facial expressions.

It is noteworthy that our system exhibits high scalability. Each internal module is meticulously designed, with training and invocation modules functioning independently. The sentiment analysis model is interchangeable, allowing the system to flexibly adapt to varying degrees of complexity in sentiment analysis tasks. In the current phase, we have opted for foundational and readily available models to implement the overall functionality. However, the system's architecture permits seamless upgrading and replacement of these models in the future.

To address issues arising from the independence of various modality inputs and constraints imposed by foundational models, the system introduces the LLMs as an independent agent. This agent is responsible for handling multi-channel sentiment inputs and generating the final analysis results. Additionally, the system employs the prompt engineering approach to inform known deficiencies in foundational models, optimizing the ultimate results and ensuring further enhancement of system performance.

After the previous processes, the system will interact the final resulting emotion in the form of natural language with the subsequent LLM modules, as shown in the structure of Figure 1.

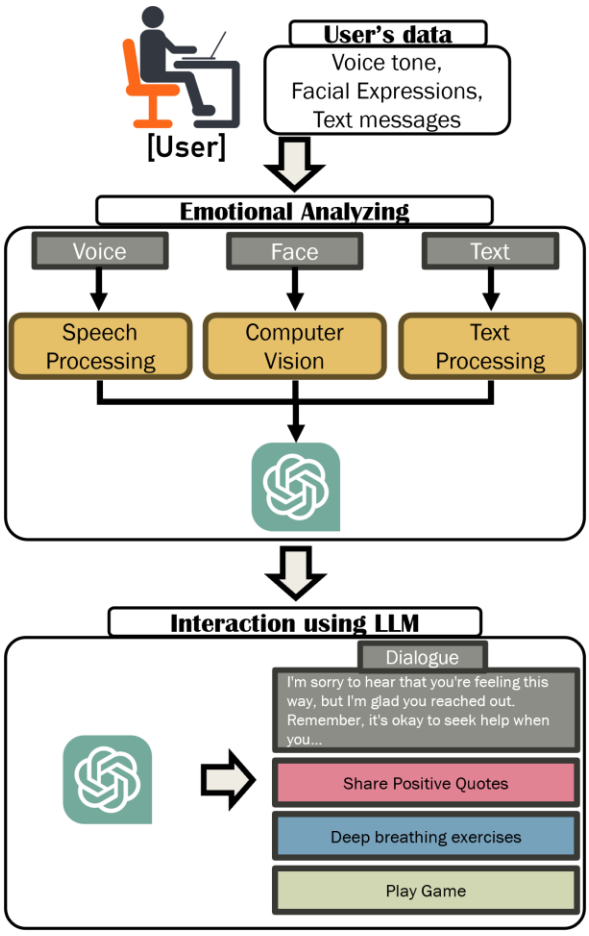

Figure 1: The whole system structure of the proposed social robot

3.2 Interaction using LLM

As mentioned earlier, there are several obstacles to using LLM such as ChatGPT for specific systems. One of them is that the output generated by LLM is not fixed and changes from time to time. Therefore, we learn LLM through prompt engineering to

obtain a stable and fixed form of output from LLM, even though the content of the conversation itself changes.

The prompt engineering structure to be used for this is divided into **Role**, **Main Task**, and **Instruction**. This can be seen in Figure 2. **Role** is designed to prescribe a specific function to the LLM. Through this element, we determine which domain of knowledge the LLM should tap into by framing a particular context of perspective. For example, by positioning the LLM as a social AI assistant for depressed aiming to care their emotional state, the LLM conceptualizes its role based on the provided context. **Main Task** is to define the primary objective for the LLM. Since the response is shaped by this directive, it is imperative that the instruction is unambiguous and succinct, utilizing domain-specific terms when necessary. In this context, the main objective is generating proper dialogue and activity suggestions based on user's emotional state. And **Instruction** contains all other information, including examples of the output form expected of LLM. Given LLM's reliance on extensive datasets as its information source, this instruction including examples imposes limitations on its actions. Based on this format, the LLM generates results in the form of fixed pythonic code.

Prompt

Role

```
{"user":"You're an AI who gets input on the emotional state of someone who's depressed."}
```

Main Task

```
{"user":"You need to generate code include a proper conversation and 3 or less suggestions of activities accordingly."}
```

Instruction

```
{"user":"Here are examples of the emotional state you will receive and the code you will need to generate accordingly. You have to follow the following format to generate the code. Other functions are not needed."}
{"user":"Example 01"}
{"user":"Sad"}
{"assistant":"
def main():
conversation("appropriate conversation for Sad human")
activities = [Activity 1, Activity 2, Activity 3]
# If your suggestion is an app, write 'app_name app' and if it is a search word, write 'search_word_name search_word'.
# And your suggested app must exist in ubuntu environments. And don't capitalize the first letter.
search_word_or_app_for_activities=[search_word_or app_name for Activity 1, search_word_or app_name for Activity 2, search_word_or app_name for Activity 3]
create_gui(activities, search_word_or_app_for_activities)
if __name__ == '__main__':
        main()
"}

{"user":"Also, Just tell me the code generated. I don't need an explanation."}
```

Figure 2: Prompts used in the prompt engineering process in this study

The final task of the Interaction using LLM part is to create a Graphical User Interface (GUI) using the pythonic code generated by LLM and prepared APIs to make it easier for users to accept the information. For this task, the output generated by LLM is divided into conversation and activities. A conversation refers to a dialogue appropriate to the user's emotional state and will be output as a voice message through the TTS API. Activities refer to activity suggestions appropriate to the user's emotional state. Depending on the type of activity, an API that enables execution of an appropriate app or search in the browser is selected, making it easy for users to execute.

4 Implementation

Based on the methodology described in Section 3, we implemented a social AI robot. The entire implementation code was written in a python script, and the code is released as an open-source repository on GitHub [8].

4.1 Emotional analyzing

For the speech emotion recognition, the system extracts features from sound using Mel-Spectrogram based on the DAVDESS dataset [9]. Due to its simplicity and cost-effectiveness in the past similar tasks, CNN [10] is chosen as the default setting to complete the system. However, in the initial design of the system, single models often perform poorly due to the high similarity between Mel-Spectrograms. Ultimately, the speech emotion recognition module is designed as a multi-agent system with 5 simple CNNs as shown in Figure 3, where each model represents an agent responsible for a simple task. Through the combination and reuse of different models, simple models are able to complete complex tasks. Finally. the system automatically records a 5-second audio snippet and generates a predicted label to store.

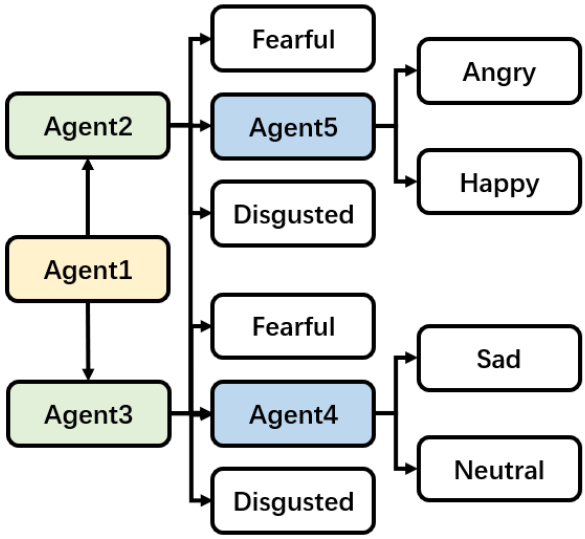

Figure 3: Multi-Agent system for Speech Emotion Recognition

Regarding the facial expression recognition, the default model of the system involves the utilization of VGG19 [11], trained on the FER2013 dataset [12]. This dataset comprises grayscale facial

images categorized into seven classes (0-6: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral). VGG19, a model commonly employed in the field of facial expression recognition, was chosen for its balance of lightweight design and robust functionality.

In the final system implementation, OpenCV is employed to recognize all facial expressions within a 5-second timeframe using the user interface as shown as Figure 4. The expression with the highest frequency is then determined as the ultimate prediction, which is subsequently stored for future reference.

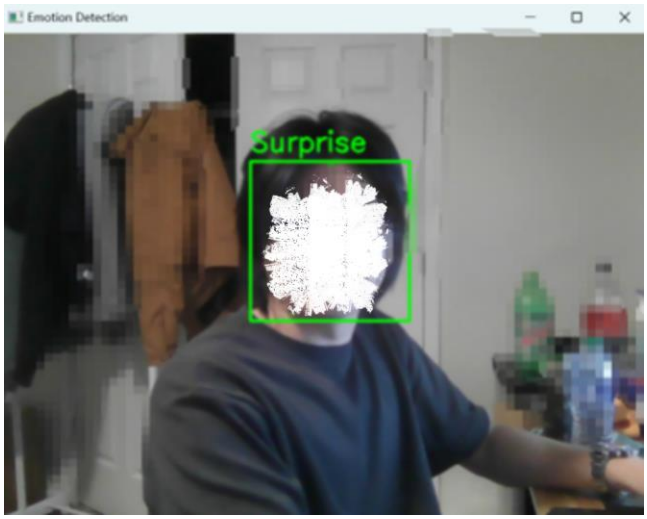

Figure 4: User Interface for facial expression recognition

When performing the text emotion recognition task, the initial model of the system was set to BERT [13], based on its superior performance in the past task. The model was fine-tuned over five epochs on a dataset [14] containing six basic emotions (happy, sad, anger, fear, love, surprise) to better fit the specific needs of the task. The system deploys a simple and easy-to-use user interface as Figure 5 to record user input and provide real-time emotion recognition. In addition, the system stores the results of each emotion recognition for subsequent analysis and reference.

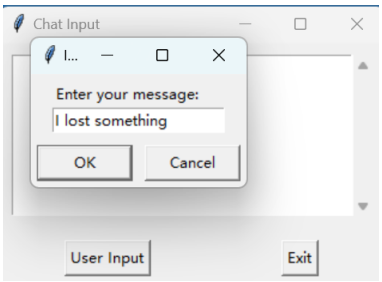

Figure 5: System structure of Interaction using LLM Part

In the end, the previous predictions will be stored together, and the LLM will be invoked as an agent to make the final judgment and it can also solve the previous problem of different module data sets with different labels, instead of the original multimodal learning process. Through prompt engineering as Figure 6, the system can be designed with manual input of "trust level" and previous base module limitations as auxiliary information, and the agent will make corresponding judgments based on the prompt. This design improves the accessibility and interpretability of the system and provides a cost-effective solution for direct deployment of multimodal models or more advanced base models.

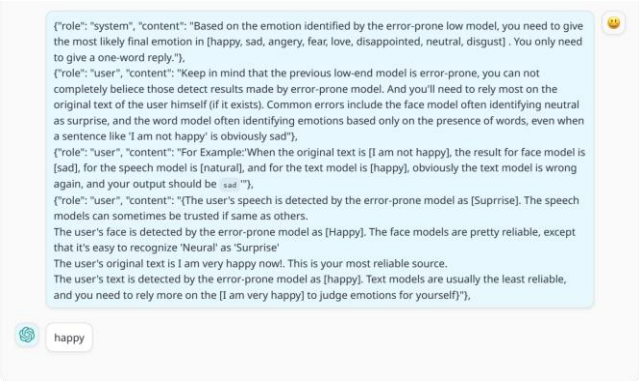

Figure 6: The LLM agent for final judgment

4.2 Interaction using LLM

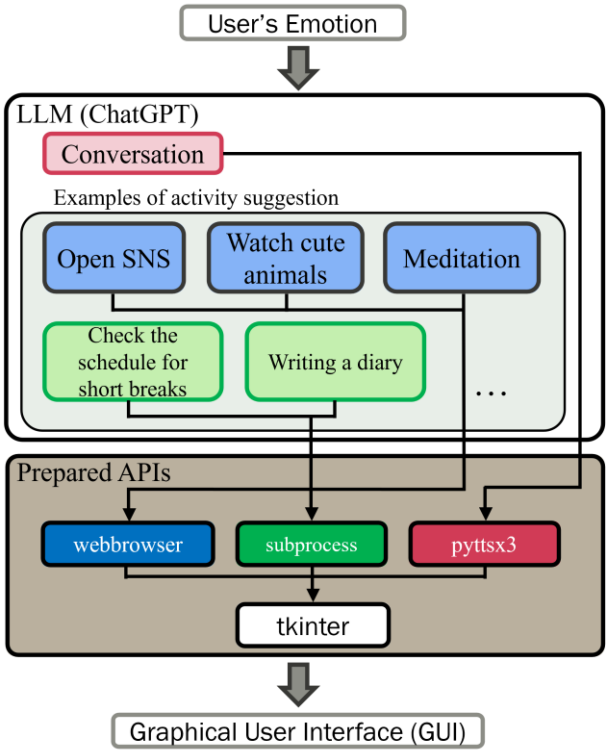

Figure 7: System structure of Interaction using LLM Part

The implementation of the Interaction using LLM part was implemented as shown in Figure 7. When the user's current emotional state is transmitted to the LLM, which has been trained

through prompt engineering to fix the output in the desired form, the LLM generates a code appropriate for user's emotional state. At this time, Open AI's ChatGPT GPT-4 model [15] was used for LLM.

The generated code is largely divided into conversation and activity suggestion. The conversation is output in the form of a voice message through the pyttsx3 API [16]. Activity suggestions are once again divided into two types, *search_word* and *app*, depending on their type. If the proposed activity is classified by *search_word*, the webbrowser API [17] automatically generates code to search for the corresponding search word in the Internet browser. If the proposed activity is classified as an *app*, code is generated to run the application by the subprocess API [18]. Lastly, the generated code is processed into a GUI for user by using the tkinter API [19]. The sample display of created GUI can be seen in Figure 8.

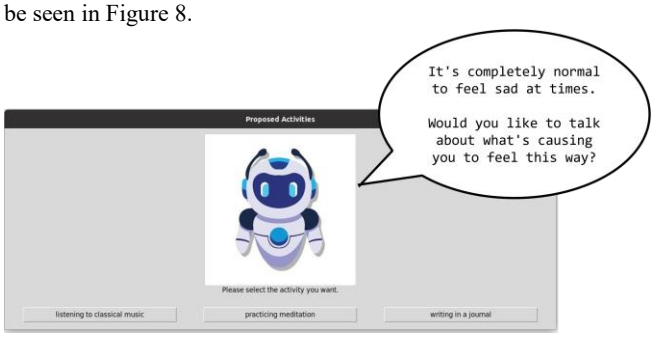

Figure 8: Sample image of GUI display

5 Results and Analysis

In this section, we describe the results and analysis of our implemented system. Analysis was conducted separately for the Emotional analyzing part and the Interaction using LLM part.

5.1 Emotional analyzing

For the Speech Emotion Recognition part, the accuracy rate is shown in Table 1. The average accuracy rate of the original five agents is 0.84, while the average accuracy rate of the final system on each task is 0.99, which is significantly improved compared with the original accuracy rate. This represents the availability of a multi-agent system architecture to achieve the effect of using a simple model to complete complex tasks.

Agent	Accuracy	Final Task	Final Accuracy
Agent 1	0.816	Happy	1.000
Agent 2	0.699	Angry	0.995
Agent 3	0.927	Neutral	1.000
Agent 4	0.894	Sad	0.995
Agent 5	0.899	Fearful	0.985
		Disgusted	0.965

Table 1: Accuracy for Speech Emotion Recognition

The confusion matrix of the final facial emotion recognition and text emotion recognition is shown in Figure 9 and the Figure 10. It was found that the facial emotion recognition performed well on some clearly featured expressions, such as happy, but there were misjudgments in the "fear" and "sad". Additional, according to the final actual test results, the data of "surprise" in the original dataset is relatively small, which also has negative effects on the practical application. Therefore, replacing the basic model while adopting a more comprehensive data set is a worthy direction to consider for improving results.

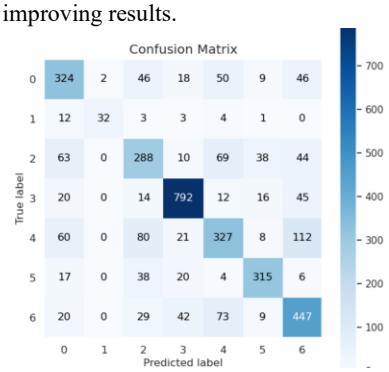

Figure 9: Accuracy for Facial Emotion Recognition

In terms of text emotion recognition, the final fine-tuning results in excellent data performance due to BERT's excellent ability and more complex design compared to other modules.

Figure 10: Accuracy for Text Emotion Recognition

Overall, the system basically meets the needs of the requirements, but it is worth noting that replacing the default model with a more recommended solution may achieve better results in marginal utility, although this also comes with higher costs and hardware requirements.

5.2 Interaction using LLM

We conducted experiments for ablation study to evaluate the implemented system. We gave each ChatGPT model a prompt with a specific structure and verified how accurately it generated the desired code. The test for each model and prompt was conducted 10 times. What LLM needs to generate is one appropriate conversation and three activity suggestions that fit the user's emotional state. Therefore, the success rate was checked to

see whether the four contents were successfully created for each test. These results can be seen in Table 2.

Model	Prompt	Success Rate (%)
GPT-3.5	Unstructured Prompt	25
	Structured Prompt	25
	Structured Prompt with additional fine-tuning	**82.5**
GPT-4	Unstructured Prompt	47.5
	Structured Prompt	**100**

Table 2: Experimental results for evaluation of Interaction using LLM part

In GPT-4 model, the desired type of code was generated with 100% accuracy when using the prompt with the proposed structure, and when using the prompt with the proposed structure and additional fine-tuning in GPT-3.5 model, the desired type of code was generated with 82.5% accuracy.

In both models, it can be seen that the structured prompt used in this report significantly raises the accuracy of LLM's output. The results from the GPT-3.5 model are also worth noting, because the GPT-3.5 model can be used for free when using the web version of ChatGPT. This is an important element in the implementation of a social AI robot that can be used by a variety of people at a low price, which is the purpose of our project.

6 Conclusions and Future Works

In this report, we introduced social AI robot system for depression. The proposed system analyzes emotions through multiple models. The emotion analysis results performed based on the text generated by the user, the user's facial expression, and the tone of the user's voice are delivered to the LLM, and the LLM integrates the results to output a single user emotional state. The user's emotional state is once again transmitted to LLM. In this process, LLM is learned to fix the output in the form of a specific pythonic code through prompt engineering. Accordingly, LLM outputs conversation and activity suggestions appropriate to the user's emotional state. This pythonic code is displayed on the screen in the form of a GUI for easy use by users based on prepared APIs.

To improve and consolidate our social AI robot's ability, we need further research. Firstly, in the part of Emotional analyzing, Multi-Agent LLM system Optimization is possible. In other words, we can explore the use of a Multi-Agent system to optimize decision-making instead of a single agent. And the current models used in this process are all basic models (e.g. BERT, VGG, CNN), which can be replaced with State of the art (SOTA) model or multi-modal emotion recognition model. This can improve the accuracy of the results. Also, in the part of Interaction using LLM, more complex interaction with users using LLM is needed. Currently, the dialogue and activity suggestions generated by LLM are too simple and one-dimensional. Although LLM generates the desired code with 100% accuracy because of this simple code structure, this can ultimately be seen as a factor that hinders the performance of the system. There is a need to improve a function of social AI robot that interacts with users in real time through additional prompts and improvements to the GUI.

ACKNOWLEDGMENTS

This project is for CNIT 58100 - Introduction to Assistive Technology and Robotics (2023 Fall) at Purdue University. In this study, Charles Li was in charge of the Emotional Analyzing part, and Taehyeon Kim was in charge of the Interaction using LLM part.

REFERENCES

[1] Reinert, M, Fritze, D. & Nguyen, T. 2022. *The State of Mental Health in America 2023*. Mental Health America, Alexandria VA.

[2] Alemi, M., Meghdari, A., Ghanbarzadeh, A., Moghadam, L. J., & Ghanbarzadeh, A. 2014. Effect of utilizing a humanoid robot as a therapy-assistant in reducing anger, anxiety, and depression. In *2014 second RSI/ISM international conference on robotics and mechatronics (ICRoM)*. Tehran, Iran, 748-753.
DOI: https://doi.org/10.1109/ICRoM.2014.6990993.

[3] Šabanović, S., Chang, W. L., Bennett, C. C., Piatt, J. A., & Hakken, D. 2015. A robot of my own: participatory design of socially assistive robots for independently living older adults diagnosed with depression. In *Human Aspects of IT for the Aged Population. Design for Aging: First International Conference, ITAP 2015*, Los Angeles, CA, USA, 104-114.
DOI: https://doi.org/10.1007/978-3-319-20892-3_11.

[4] Chen, S. C., Jones, C., & Moyle, W. 2018. Social robots for depression in older adults: a systematic review. *Journal of Nursing Scholarship*, 50(6), 612-622.
DOI: https://doi.org/10.1111/jnu.12423.

[5] Lee, H. R., Šabanović, S., Chang, W. L., Nagata, S., Piatt, J., Bennett, C., & Hakken, D. 2017. Steps toward participatory design of social robots: mutual learning with older adults with depression. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*. New York, NY, USA, 244-253.
DOI: https://doi.org/10.1145/2909824.3020237.

[6] Abdollahi, H., Mollahosseini, A., Lane, J. T., & Mahoor, M. H. 2017. A pilot study on using an intelligent life-like robot as a companion for elderly individuals with dementia and depression. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. Birmingham, UK, 541-546.
DOI: https://doi.org/10.1109/HUMANOIDS.2017.8246925.

[7] Pandey, A. K., & Gelin, R. 2018. A mass-produced sociable humanoid robot: Pepper: The first machine of its kind. *IEEE Robotics & Automation Magazine*, 25(3), 40-48.
DOI: https://doi.org/10.1109/MRA.2018.2833157.

[8] Charles Li & Taehyeon Kim. 2023. socialbot. (December 2023). Retrieved December 8, 2023 from https://github.com/Charles-HC-Li/socialbot.

[9] Steven R Livingstone & Frank A Russo. 2018. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE*, 13, 5, 1-35.
DOI: https://doi.org/10.1371/journal.pone.0196391.

[10] Kiron O'Shea & Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. arXiv.org. Retrieved December 12, 2023 from https://arxiv.org/abs/1511.08458.

[11] Karen Simonyan & Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv.org. Retrieved December 12, 2023 from https://arxiv.org/abs/1409.1556.

[12] Manas Sambare. 2013. FER-2013. Kaggle.com. Retrieved December 12, 2023 from https://www.kaggle.com/datasets/msambare/fer2013.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.org. Retrieved December 12, 2023 from https://arxiv.org/abs/1810.04805.

[14] Emotion_final.csv at main · hamziqureshi/Emotion_Classification_with_BERT. GitHub. (Mar 2022) Retrieved December 12, 2023 from https://github.com/hamziqureshi/Emotion_Classification_with_BERT/blob/main/Emotion_final.csv.

[15] OpenAI. 2023. ChatGPT (Nov 30 version). (December 2023). Retrieved December 8, 2023 from https://chat.openai.com.

[16] Natesh Bhat. 2017. pyttsx3. (December 2023). Retrieved December 8, 2023 from https://github.com/nateshmbhat/pyttsx3.
[17] Python Software Foundation. 2002. webbrowser. (October 2023). Retrieved December 8, 2023 from https://docs.python.org/3/library/webbrowser.html#module-webbrowser.
[18] Python Software Foundation. 2015. subprocess. (October 2023). Retrieved December 8, 2023 from https://docs.python.org/3/library/subprocess.html.
[19] Lundh, F. 1999. An introduction to tkinter. (December 2023). Retrieved December 8, 2023 from https://docs.python.org/3/library/tkinter.html.