Harnessing NLP for Real-World Impact on Mental Health

Sarthak Jain
University of Illinois at Chicago
sjain217@uic.edu

Abstract

Mental health is becoming an increasingly important issue nowadays. People of all ages are battling mental health problems. Developing automated systems that can help in detection and rehabilitation is crucial given the increase in caseload. In this report, we build upon an existing NLP based method that can help in rehabilitation of mental health patients. The existing method automates the Social Skills Performance Assessment (SSPA) task by developing a model that can conduct the interview and subsequently score them. The current model that conducts the interview with the patients does not recognize the 2 separate tasks that it is expected to perform, rather it just learns the existing patient utterances. We address this limitation of the existing method by formulating the interview generation problem as two separate role-playing scenarios. We conduct experiments with two separate decoder only models, using automatic evaluation metrics to evaluate the generated responses. We obtain better results for one of the decoder-only models when compared to the existing T5-base model. We conclude that incorporating role-playing techniques in the existing T5 base model and increasing model sizes may improve the interview generation performance. Although the method we propose is general, we focus on two mental health conditions (MHCs): Schizophrenia, and Bipolar Disorder.

1 Introduction

Note: The project described in this report is a continuation of the work done by Ankit Aich in the NLP lab at the University of Illinois at Chicago. Thus, the content of this section draws heavily from his thesis report.

The applications of AI are huge in the field of healthcare in general. In fact, since 2010, when the FDA decided to approve the use of AI in healthcare, there has been an increase in the usage of AI based medical medical devices (Zhu et al., 2022). However, NLP based technologies are not being deployed extensively. This is not true for other areas of AI such as Computer Vision. For example, studies such as (Mokoatle et al., 2023)(Esteva et al., 2021)(Dai et al., 2021) show that computer vision research is being successfully deployed in healthcare areas such as cancer detection and diabetic retinopathy. So, the major question now is why are NLP systems not being deployed for everyday usage in healthcare.

A first guess that anyone might make towards the lack of deployable NLP applications is a lack of interest in developing such systems. However, the reason is far from lack of interest as there are survey papers such as (Valizadeh and Parde, 2022) that report a myriad of papers in the domain. There have also been Language models such as (Ji et al., 2022) that have been created specifically for use in healthcare applications.

To understand the reason behind a lack of deployable NLP systems, a systematic review of the papers from ACL anthology was conducted when the project began. The following two major trends emerged that explain why NLP systems are not being used in healthcare:

1. **Heavy Usage of Social Media Data:** There have been many works that utilize social media data as their dataset to perform further analysis. The medical condition in these posts is generally self-disclosed or identified by an expert clinician. For example, in (Birnbaum et al., 2017) clinical experts verified the self-disclosure of Schizophrenia by users and then built classifiers to classify new tweets from users being schizophrenic or not. On the other hand, in (Pedersen, 2015) the self-disclosure of depression in tweets was considered as the gold standard. Utilizing social media datasets for building NLP systems has many pitfalls. First, the veracity of self-disclosure in social

media posts cannot be trusted. Even if the condition in the posts are verified by a clinician like in (Birnbaum et al., 2017) it is only the post that is being used to identify the condition of the user. Second, the social media post is not a clinical setting. The language used in the posts is colloquial and rapidly changing. Using just a few words to label the medical condition of a user does not follow established medical procedures, thus the diagnosis cannot be trusted.

2. **Lack of domain flow in methods:** Most papers in the past have failed to successfully incorporate clinical knowledge in the NLP pipeline. For example, papers such as (Sherman et al., 2021) (Amir et al., 2019) (Aguirre and Dredze, 2021) do not focus on clinical knowledge when building the NLP pipeline. Others such as (Fraser et al., 2016) follow a method-first-diagnosis-second approach. In such an approach a dataset is first labeled with the medical condition. Subsequently, generic ML algorithms are used to classify each example in a dataset into the datset labels. Such an approach is flawed because there is a disconnect between the data and methods: the methods used for classification are not clinically verified and thus cannot be trusted by medical practitioners. For example, (Fraser et al., 2016) use part-of-speech tags, richness and lexical diversity to find depressed users, These features have not been clinically proven to be useful for diagnosis of depression and thus are not very useful to medical experts. Rather, a much better approach is to use ML algorithms to find patterns of speech in diagnosed users or to help in their rehabilitation, thus leaving the diagnosis solely on medical community.

The result of these trends has been that the methods and data described in the papers that use NLP techniques has not been clear on how they help in understanding and advancing the field further. Consequently the methods have not been used by the medical community.

This project aims to precisely address these limitations. We utilize data collected from real patients through a process called the SSPA (Patterson et al., 2001), and utilize methods that have been clinically verified, thus incorporating domain flow in methods. SSPA consists of conducting interviews with patients in two different scenarios, and scoring the interviews on parameters that are descriptive of social skills of the patients. In both the scenarios, a medical professional assumes the role of an interviewer role-playing the character specific to the scenario. The following are the two scenarios that interviewer and the patient act out:

- The first scenario consists of the interviewer and patient conversing as neighbors. The scene is that the interviewer has recently moved in as a neighbor of the patient, and the two are talking about general aspects of the neighborhood such as safety, nature of neighbors, etc. The aim here is to assess the social skills of the patient in a friendly setting.

- The second scenario is a confrontational setting in which the interviewer acts as the landlord and the patient as the tenant. The scene is that the patient had complained to the interviewer about a prior leak in the apartment but it has not been fixed yet. The patient confronts the landlord about the unfixed leak. The aim here is to assess the social skills of the patient in a confrontational setting.

The interviews follow an outlined procedure in accordance with the medical guidelines.

In this project, we aim to improve the existing method that automates the SSPA process by building a model that can conduct the interviews and subsequently score them as well. Specifically we aim to improve the interview generation model. Currently, the model simply learns to mimic interviewer responses from the given data without understanding the SSPA scenarios. We address this limitation by formulating the interview process as two separate role-playing scenarios and making the model mimic the role in each scenario, thus making the model well aware of the situation in the SSPA process. We experiment with two separate decoder-only transformer (Vaswani et al., 2023) models and we obtain better results for one of the decoder models than the results of the existing T5-base (Raffel et al., 2020) model.

The rest of the report is organized as follows:

- Section 2 provides a review of the existing work on role-playing LLMs.

- Section 3 presents the method we use for our problem

- Section 4 presents the experimental details and results
- Section 5 presents the possible directions that can extend this work.
- Section 6 concludes the work presented in this report.

2 Previous Work

In this section, we review the literature that discusses role-playing LLMs. Section 2.1 discusses the inherent nature of every LLM. Section 2.2 discusses one method of making LLMs role-play by prompting them with character information. Finally, Section 2.3 talks about a popular paper of making LLMs role-play a character.

2.1 Every LLM Role-plays Characters

In this section, we will discuss the nature of responses given out by any LLM trained on public data (before fine-tuning). (Shanahan et al., 2023) talks about the behavior of LLMs when they are used as dialogue agents. The authors show how people who are unaware of how LLMs are trained may ascribe human characteristics to LLMs. For example, they may believe that the LLM based dialogue agent "knows", "thinks" or "understands". However, ascribing such characteristics to LLMs is a mistake since they are trained on public data, and, just say what the training data says. Specifically, the LLM responses can be thought of as a superposition of the characters that they are trained on. So as (Willison) talks about cases where the Bing chatbot acts rudely and starts threatening users, it actually does not have any such beliefs, rather it is role-playing a character that is rude or threatens users.

2.2 Prompting LLM with character information

As the Section 2.1 points out, since the responses of an LLM not fine-tuned on a specific dataset is a superposition of characters, we need techniques that can make LLM responses consistent and make it role-play a specific character. We will discuss one of those techniques in this section.

The most straightforward approach to make an LLM role-play a character in a conversation setting is to fine-tune a decoder only model to make it generate the character responses. The input to such a model is a prompt that contains the information about the character and a part of conversation just before the character's response. The expected output is the character's response. Methods utilizing such an approach differ in the way they construct their prompts. We describe two methods here following this approach:

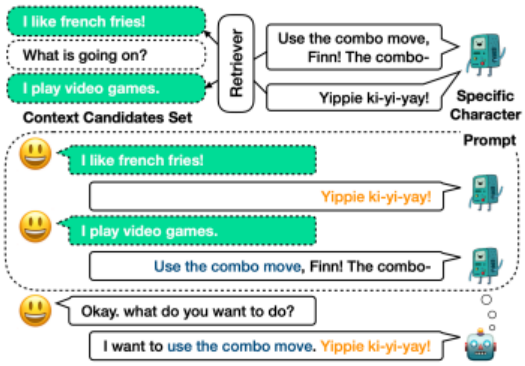

Figure 1: Illustration of method described in Section 2.2.1. The utterances "Use the combo move.." and "Yippie ki-yi..." are the available character utterances. The utterances "I like french fries!", "What is going on?", and "I play video games." are the utterances in the candidate set C. The box labeled prompt is what is fed to the model as the input prompt along with the query "Okay. what do you want to do?"

2.2.1 Single Turn Role-play

(Han et al., 2022) introduce a very rudimentary approach which is designed to mimick a specific character's utterance in response to a given query. Thus, this approach is not designed to role-play a character in a multi-turn conversation, rather it mimicks a character's response to a specific query. The method is given a set of character's utterances, and a candidate context set C. For each given character utterance, a retriever R is used that selects the most appropriate candidate query from the set C. This gives a set T, where each element is a pair of the character's utterance and the candidate query matched by the retriever R. This set is fed to a decoder-only transformer LLM as a prompt in a dialogue format along with the query q to which the character's response is expected. The point to note here is that there is no fine-tuning performed, rather the method utilizes the in-context learning capability of LLMs to learn from the dialogue-formatted prompt and generate the character's response. Figure 1 illustrates the method.

2.2.2 ChatHaruhi

(Li et al., 2023) introduce a more nuanced tech-

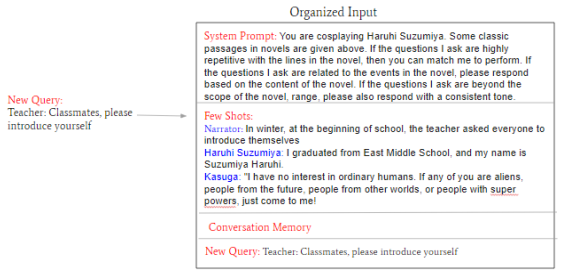

Figure 2: Figure illustrating the prompt construction as described in Section 2.2.2

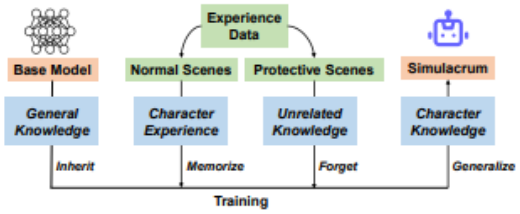

Figure 3: Figure illustrating the Character-LLM method introduced in Section 2.3.

nique for making an LLM role-play in a multi-turn conversation setting. The idea is to embed enough information in the prompt itself so as to make the LLM understand the scenario. Specifically, a decoder-only transformer LLM is fed a prompt which consists of the following 4 components:

- **System Prompt:** This component informs the LLM of the scenario at hand, the character the LLM is expected to role-play, information about the character and any other information that may be useful to introduce the situation.
- **Character Responses:** The methodology introduced in the paper is designed to mimic characters from novels, movies or TV-shows. These characters have dialogues peculiar to them. Thus, this component of the prompt contains classic responses of the characters to common queries.
- **Conversation Memory:** A character's response to a query not only depends on the preceding dialogue but also on the conversation history of the ongoing conversation. Thus, this component contains the conversation history to make the responses of the LLM coherent and relevant to the ongoing conversation.
- **New Query:** This component contains the new utterance of the user to which the LLM is expected to respond.

The LLM is then fine-tuned on the character's responses. Figure 2 illustrates the prompt construction for an example user query.

2.3 Character-LLM

(Shao et al., 2023) introduce a significantly different approach than the one presented in Section 2.2 to mimick a character. The approach is called as Character-LLM. Specifically, the motivation of the approach is that the responses of a person in a conversation are based on the experiences accumulated during their lifetime. Thus, instead of making an LLM just learn their utterances, if we are able to make them learn their thought process and experiences, then the responses of the LLM will better reflect the character's personality. To that end, they first collect the experiences of the character from a trusted source, and then make an LLM learn those experiences. The drawback of this method is that a separate model is required for each character that the LLM mimicks. Figure 3 illustrates the workflow of the method.

3 Method

In this section, we will describe the method we have used to make an LLM roleplay as the neighbor or landlord in the two scenarios described in Section 1. As described in Section 2.2.1, the method is only for single turn conversation. Since our application is of multi-turn conversation, we do not utilize that method. On the other hand, although the method described in Section 2.3 can be utilized for multi-turn conversations, it requires a separate model for each character. Moreover, in our case we do not have the experiences of either the landlord or the neighbor that the LLM is expected to role-play. Thus, we refrain from utilizing this method as well.

However, the method described in section 2.2.2 perfectly fits our use-case: we have two separate characters (as described in Section 1) that need to be mimicked in a multi-turn conversation using a single LLM. Thus we use that method. As described in that section, since the uniqueness of the method lies in the prompt that is constructed, we adapt the prompt described in the section to our use-case. Specifically, we modify the System Prompt component while doing away with the Few Shots Component. The remaining two components

remain the same. We utilize the following system prompts for the two scenarios:

1. **Neighbor Scenario:** The prompt we use for this scenario is the following: *I want you to roleplay a neighbor who has moved in to a new neighborhood. If your neighbor's utterance are related to the utterance you have already seen, please try to reuse the original response lines.*

2. **Landlord Scenario:** The prompt we use for this scenario is the following: *I want you to roleplay a landlord who is conversing with a tenant. If your tenant's utterance are related to the utterance you have already seen, please try to reuse the original response lines. The situation is that tenant had complained to you about a leak earlier but that has not been fixed yet.*

These prompts are fed to a decoder-only transformer LLM and the target utterance is the interviewer's response to the query component of the prompt. The LLM is then-finetuned to respond either as the landlord or neighbor. The good thing here is that we can make a single model role-play both neighbor and landlord.

4 Experiments & Results

In this section, we will present the experimental setup and the results. Section 4.1 presents the experimental setup, Section 4.2 presents the metrics used for evaluation, Section 4.3 presents the results, and Section 4.4 presents a discussion on the obtained results.

4.1 Experimental Setup

In this section we will present the details on the setup we used to conduct the experiments. As stated in Section 1, this project is an extension of an ongoing project, so we already have the results of the T5-base (Raffel et al., 2020) model.

The dataset we have is the transcripts of the conversations between the interviewer and patients(for the two scenarios described in Section 1). The transcripts were converted to txt format from audio files by a trusted third party service. The format of txt files is shown in Figure 4. We preprocessed these transcripts to extract the interviewer and patient responses. The patient responses were extracted by searching for the string *"Patient:"*, and interviewer responses were extracted by searching for the string *"Interviewer:"*. In total we have 1100 conversation transcripts covering the three cases of Bipolar Disorder, Healthy Control, and Schizophrenia.

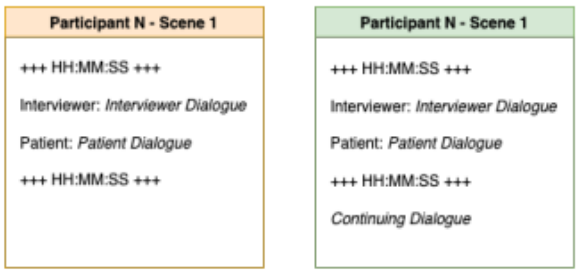

Figure 4: Transcription formats prior to preprocessing. The format at right was used when patient or interviewer utterances exceeded a given timestamp and continued onward into the next dialogue block.

In our project we evaluate two more models, both of which are decoder-only transformer models. The first model, called DialoGPT (Zhang et al., 2020b) is based on the GPT-2 (Radford et al., 2019) model. DialoGPT is available in three variants: DialoGPT-small, DialoGPT-medium, and DialoGPT-large. They differ in the number of parameters and have 120M, 350M, and 770M parameters respectively. The second model, called TinyLlama-1.1B (Zhang et al., 2024) is based on the LlaMa model (Touvron et al., 2023). Both DialoGPT and TinyLlama have been trained on human conversation which means that the models already have the knowledge of the properties of human conversations. Please refer to section 3 to find the details of the input and the output of the model. When feeding the input to the model, we separate the system prompt and each turn in the conversation by an eos token. For example, an input for the neighbor scenario may look like this:

<s> I want you to roleplay a neighbor who has moved in to a new neighborhood. If your neighbor's utterance are related to the utterance you have already seen, please try to reuse the original response lines.</s><s> Hi, are you the new— H-Have you just moved in?</s>

The models were fine-tuned on a single A100 GPU for 5 epochs. We evaluated DialoGPT-small, DialoGPT-large, and TinyLlama-1.1B, thus excluding DialoGPT-medium from our evaluation. Table 1 lists the training time of the DialoGPT and TinyLlama models.

Model	Time(s)
DialoGPT-small	2567.07
DialoGPT-Large	13644.87
TinyLlama-1.1B	19237

Table 1: Table listing the training times of the models

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.04	0.05	0.04	0.04	0.05	0.04	0.15	0.15	0.15
BD Scene2	0.13	0.14	0.13	0.13	0.14	0.12	0.26	0.25	0.26
HC Scene1	0.04	0.06	0.05	0.04	0.06	0.05	0.14	0.14	0.14
HC Scene2	0.11	0.12	0.13	0.11	0.12	0.13	0.23	0.23	0.23
SZ Scene1	0.04	0.06	0.04	0.04	0.06	0.04	0.16	0.15	0.16
SZ Scene2	0.12	0.15	0.12	0.12	0.15	0.12	0.26	0.25	0.25

Table 2: DialoGPT-small results

4.2 Evaluation Metrics

To evaluate the model performance, we need to evaluate both the syntactic and semantic similarity of the generated response to the original interviewer response. To evaluate the syntactic similarity, we use the Rouge1,and RougeL variants of Rouge Score(Lin, 2004). To evaluate the semantic similarity, we use the BertScore(Zhang et al., 2020a).

4.3 Results

Table 2, Table 3, Table 4, and Table 5 give the results for DialoGPT-small, DialoGPT-large, TinyLlama-1.1B, and T5 respectively. In the tables, BD Scene 1 and BD Scene 2 denote Bipolar Disorder Scene 1 and Scene 2 respectively. HC Scene 1 and HC Scene 2 denote Healthy Control Scene 1 and Scene 2 respectively. Finally, SZ Scene 1 and SZ Scene 2 denote Schizophrenia Scene 1 and Schizophrenia Scene 2 respectively. Scene 1 refers to the Neighbor scene and Scene 2 refers to the landlord scene. The observations from these results are presented in the next section.

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.15	0.21	0.15	0.15	0.20	0.15	0.45	0.42	0.43
BD Scene2	0.23	0.28	0.24	0.23	0.27	0.24	0.56	0.54	0.55
HC Scene1	0.14	0.18	0.15	0.14	0.18	0.15	0.42	0.40	0.41
HC Scene2	0.26	0.29	0.27	0.25	0.28	0.27	0.57	0.56	0.57
SZ Scene1	0.13	0.17	0.14	0.13	0.17	0.13	0.44	0.42	0.43
SZ Scene2	0.26	0.30	0.26	0.25	0.29	0.26	0.57	0.55	0.56

Table 3: DialoGPT-Large Results

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.22	0.28	0.24	0.21	0.27	0.24	0.74	0.73	0.73
BD Scene2	0.36	0.39	0.39	0.35	0.37	0.37	0.78	0.77	0.77
HC Scene1	0.27	0.32	0.31	0.27	0.31	0.30	0.76	0.75	0.75
HC Scene2	0.36	0.38	0.38	0.35	0.37	0.37	0.77	0.77	0.77
SZ Scene1	0.23	0.28	0.26	0.22	0.27	0.25	0.75	0.73	0.74
SZ Scene2	0.36	0.39	0.39	0.35	0.38	0.37	0.77	0.76	0.77

Table 4: TinyLlama-1.1B Results

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.36	0.38	0.38	0.34	0.36	0.37	0.66	0.66	0.66
BD Scene2	0.33	0.36	0.34	0.31	0.34	0.33	0.61	0.61	0.61
HC Scene1	0.36	0.39	0.39	0.36	0.38	0.39	0.67	0.68	0.67
HC Scene2	0.38	0.40	0.39	0.37	0.39	0.38	0.64	0.64	0.63
SZ Scene1	0.30	0.33	0.31	0.30	0.32	0.31	0.63	0.64	0.63
SZ Scene2	0.34	0.37	0.36	0.34	0.36	0.35	0.62	0.63	0.61

Table 5: T5 results

4.4 Observations

We can make the following observations from the results presented in the previous section:

1. From Table 2, Table 3, and Table 4, we see that both Rouge score and BertScore improve with increase in the decoder model size. This suggests that, for a given architecture, as the model size increases, the model performance also improves.

2. TinyLlama gives the best BertScore.

3. TinyLlama and T5 give similar Rouge score.

4. The size of T5 is only 200M, but its results are only slightly worse than TinyLlama. This suggests that the encoder-decoder architecture may be better suited for this task. By incorporating role-playing strategies within the T5 architecture, we may obtain better results.

5 Future Work

Based on the results obtained in Section 4.3, the following future directions seem promising:

1. Currently, T5 is fine-tuned without incorporating role-playing strategies. As the results show, incorporating role-playing strategies may improve its performance. Thus, making T5 role-play the neighbor and landlord is a good next step.

2. As we saw from the results, increasing the model size improved the model performance, thus increasing the model size is a promising next step. Since increasing the model size may make it difficult to load the model on a single GPU, multi-GPU training would be required.

3. It is common knowledge that LLMs are sensitive to the input prompts. Thus, trying out different prompts may improve the model performance further.

4. Currently, the models are evaluated using automatic evaluation metrics. However, since the models are expected to be deployed for use with real patients, it is necessary to conduct human evaluation to get a complete picture.

6 Conclusion

There is huge potential for use of AI technologies in the medical field. However, unlike other fields of AI, NLP technologies are not being used much in real world. There are 2 main reasons for this: (1) Heavy use of social media data when building NLP based methods, and (2) Lack of domain informed NLP methods. To address these shortcomings, we have used real data from real patients and incorporated the knowldge of SSPA process in our NLP pipeline. We have specifically focused on the interview generation process, modeling the problem as role-playing LLMs. We experimented with 2 decoder-only transformer LLMs: DialoGPT and TinyLlama, and found that TinyLlama gave the best BertScore, while giving comparable Rouge Score to T5-base. This suggests that either increasing the decoder architecture model size, or incorporating role-playing strategies in an encoder-decoder architecture may improve the existing results.

References

Carlos Aguirre and Mark Dredze. 2021. Qualitative analysis of depression models by demographics. In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 169–180, Online. Association for Computational Linguistics.

Silvio Amir, Mark Dredze, and John W. Ayers. 2019. Mental health surveillance over social media with digital cohorts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 114–120, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael L Birnbaum, Sindhu Kiranmai Ernala, Asra F Rizvi, Munmun De Choudhury, and John M Kane. 2017. A collaborative approach to identifying social media markers of schizophrenia by employing machine learning and clinical appraisals. *J Med Internet Res*, 19(8):e289.

Ling Dai, Liang Wu, Huating Li, Chun Cai, Qiang Wu, Hongyu Kong, Ruhan Liu, Xiangning Wang, Xuhong Hou, Yuexing Liu, Xiaoxue Long, Yang Wen, Lina Lu, Yaxin Shen, Yan Chen, Dinggang Shen, Xiaokang Yang, Haidong Zou, Bin Sheng, and Weiping Jia. 2021. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature Communications*, 12(1):3242.

Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. 2021. Deep learning-enabled medical computer vision. *npj Digital Medicine*, 4(1):5.

Kathleen C. Fraser, Frank Rudzicz, and Graeme Hirst. 2016. Detecting late-life depression in Alzheimer's disease through analysis of speech and language. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, pages 1–11, San Diego, CA, USA. Association for Computational Linguistics.

Seungju Han, Beomsu Kim, Jin Yong Yoo, Seokjun Seo, Sangbum Kim, Enkhbayar Erdenee, and Buru Chang. 2022. Meet your favorite character: Open-domain chatbot mimicking fictional characters with only a few utterances. *Preprint*, arXiv:2204.10825.

Shaoxiong Ji, Tianlin Zhang, Luna Ansari, Jie Fu, Prayag Tiwari, and Erik Cambria. 2022. MentalBERT: Publicly available pretrained language models for mental healthcare. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7184–7190, Marseille, France. European Language Resources Association.

Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. 2023. Chatharuhi: Reviving anime character in reality via large language model. *Preprint*, arXiv:2308.09597.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Mpho Mokoatle, Vukosi Marivate, Darlington Mapiye, Riana Bornman, and Vanessa. M. Hayes. 2023. A review and comparative study of cancer detection using machine learning: Sbert and simcse application. *BMC Bioinformatics*, 24(1):112.

Thomas L Patterson, Sherry Moscona, Christine L McKibbin, Kevin Davidson, and Dilip V Jeste. 2001. Social skills performance assessment among older patients with schizophrenia. *Schizophrenia Research*, 48(2):351–360.

Ted Pedersen. 2015. Screening Twitter users for depression and PTSD with lexical decision lists. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 46–53, Denver, Colorado. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role-play with large language models. *Preprint*, arXiv:2305.16367.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *Preprint*, arXiv:2310.10158.

Eli Sherman, Keith Harrigian, Carlos Aguirre, and Mark Dredze. 2021. Towards understanding the role of gender in deploying social media-based mental health surveillance models. In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 217–223, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Mina Valizadeh and Natalie Parde. 2022. The AI doctor is in: A survey of task-oriented dialogue systems for healthcare applications. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6638–6660, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Simon Willison. Bing: "i will not harm you unless you harm me first". `https://simonwillison.net/2023/Feb/15/bing/`.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. Dialogpt: Large-scale generative pre-training for conversational response generation. *Preprint*, arXiv:1911.00536.

Simeng Zhu, Marissa Gilbert, Indrin Chetty, and Farzan Siddiqui. 2022. The 2021 landscape of fda-approved artificial intelligence/machine learning-enabled medical devices: An analysis of the characteristics and intended use. *International Journal of Medical Informatics*, 165:104828.