Harnessing NLP for Real-World Impact on Mental Health

Sarthak Jain
MS Computer Science, Summer 2024
University of Illinois at Chicago
UIN: 673197580

Primary Advisor:
Prof. Natalie Parde

Secondary Advisor:
Prof. Sourav Medya

Contents

Harnessing NLP for Real-World Impact on Mental Health

July 30, 2024

Abstract

Mental health is becoming an increasingly important issue nowadays. Developing automated systems that can help in detection and rehabilitation is crucial given the increase in caseload. In this report, we focus on two mental health conditions (MHCs): Schizophrenia, and Bipolar Disorder. We iteratively refine an approach to automatically conduct and score Social Skills Performance Assessment (SSPA) interviews, to aid in the rehabilitation of the patients. The existing approach for conducting the interviews does not recognize the two separate tasks that it is expected to perform, rather it just learns the existing patient utterances. We address this limitation by formulating the interview generation problem as two separate role-playing scenarios. We conduct experiments with four different transformer based models, using automatic evaluation metrics to evaluate the generated responses. We obtain the best results for TinyLlama1.1B-Chat model. We also present preliminary human evaluation results of the model.

1 Introduction

The applications of AI are huge in the field of healthcare in general. 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.

There are two major reasons why NLP systems are not being used in healthcare. First, many existing works utilize social media data to perform further analysis. The medical condition in these posts is generally self-disclosed or identified by an expert clinician. This approach has the pitfalls of untrustworthy self-disclosures, and the social media posts being insufficient to reach a diagnosis. Second, most works, for example [Sherman et al., 2021] [Amir et al., 2019] [Aguirre and Dredze, 2021] fail to incorporate clinical knowledge in their NLP pipeline. They follow a method-first-diagnosis-second approach in which generic ML algorithms are used to classify examples from a dataset labeled for a medical condition. Such methods are not very useful because they are not clinically verified and thus cannot be trusted by medical practitioners.

The result of these trends has been that the methods and data described in the papers that use NLP techniques have 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, and utilize methods that have been clinically verified. The data is collected through SSPA [Patterson et al., 2001] interviews. SSPA consists of an interviewer 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.

Currently, the SSPA interviews are administered by human clinicians with heavy case loads[Coombs et al., 2021]. So, it is necessary to develop methods that can automate the interview process. Currently there is a system that can conduct interviews with the patients and subsequently score them. In this project, we aim to improve the interview generation model that conducts the interviews with the patients. 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 both encoder-decoder and decoder-only transformer [Vaswani et al., 2023] architectures. We obtain the best results for TinyLlama1.1B-chat model when evaluated on automatic metrics. We also present preliminary human evaluation results of TinyLlama1.1B-Chat model.

The rest of the report is organized as follows: Section 2 provides a review of the existing work on role-playing LLMs, and describes the SSPA process. 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 and also describe the SSPA interview process. 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. Section 2.3 talks about a popular paper of making LLMs role-play a character. Finally, Section 2.4 describes the SSPA interview process.

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] talk 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:

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

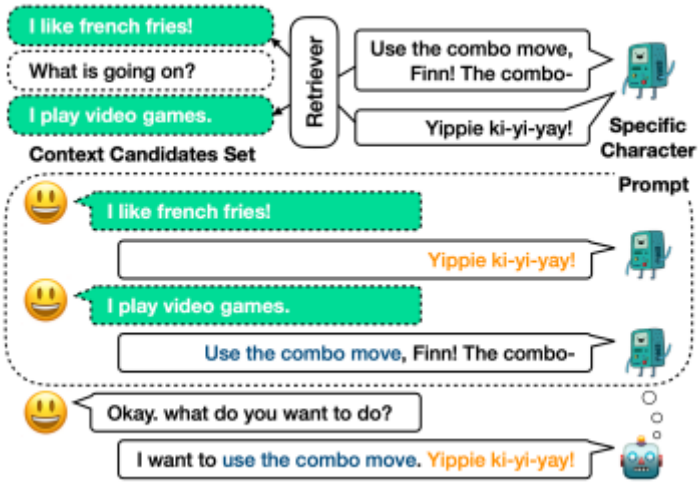

Figure 1: Illustration of method described in Section 2.2.1. The utterances "Use the combo move.." and "Yippie ki-yi..." are the available character utterances. The utterances "I like french fries!", "What is going on?", and "I play video games." are the utterances in the candidate set C. The box labeled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

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 technique 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 introduces the situation at hand such as the scenario and character information.
- **Character Response:** 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:** This component contains the conversation history of the ongoing conversation 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

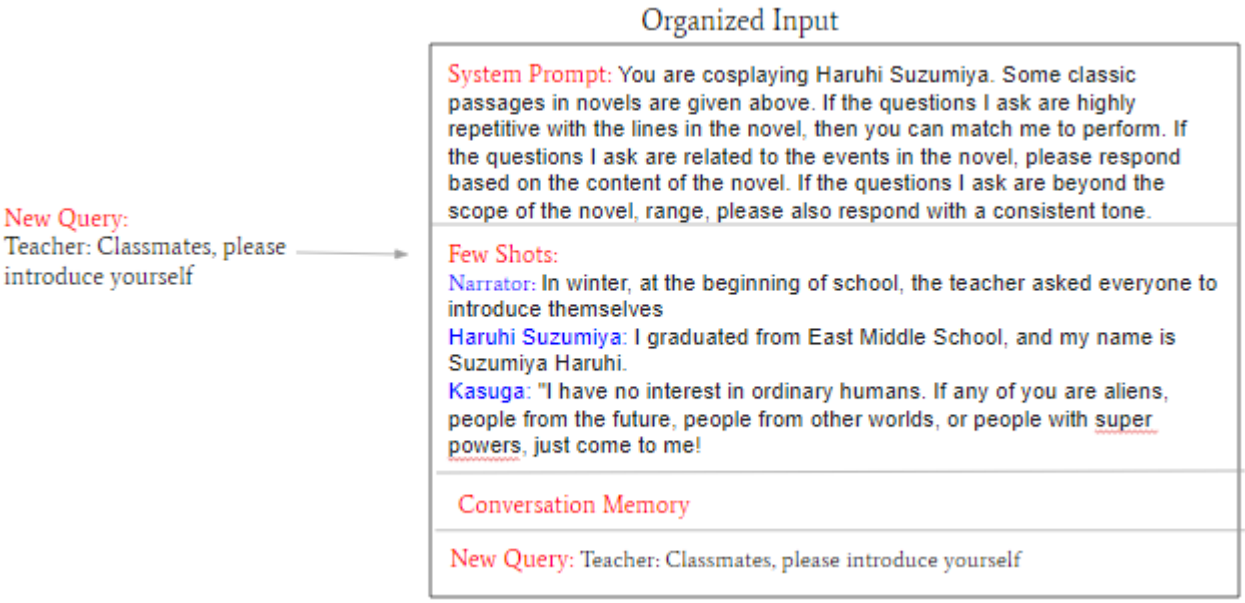

Figure 2: Figure illustrating the prompt construction as described in Section 2.2.2

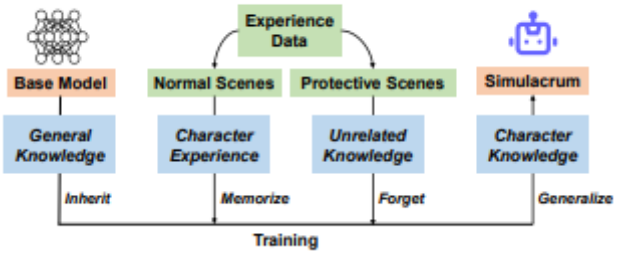

Figure 3: Figure illustrating the Character-LLM method introduced in Section 2.3.

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.

2.4 SSPA Interview Process

SSPA [Patterson et al., 2001] is a test designed to assess the social skills of a person in two different situations. The two situations are role-play scenarios, each lasting three minutes, acted out between an interviewer (a medical professional) and the subject. The scenarios are audio-recorded for subsequent scoring. The two scenarios are as follows:

- Scene 1: In this scenario, the subject plays the role of a friendly neighbor greeting the interviewer. The aim here is to assess the social skills of the subject in a friendly setting.
- Scene 2: In this scenario, the subject plays the role of a tenant calling the landlord about a leak that has previously gone unfixed. The aim here is to assess the social skills of the subject in a confrontational setting.

In both the scenarios, the interviewer tries to reciprocate the conversation initiated by the subject without leading them with additional questions. However, when the subject does not speak for atleast 10 seconds, the interviewer prompts the subject to keep the conversation going.

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 [Han et al., 2022] 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 Character Response 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. In all the tables presented in this section, Scene 1 refers to the Neighbor scene and Scene 2 refers to the landlord scene. 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.

4.1.1 Dataset

The dataset we have consists of the transcripts of the conversations between the interviewer and patients(for the two scenarios described in Section 2.4). The transcripts were converted to txt format

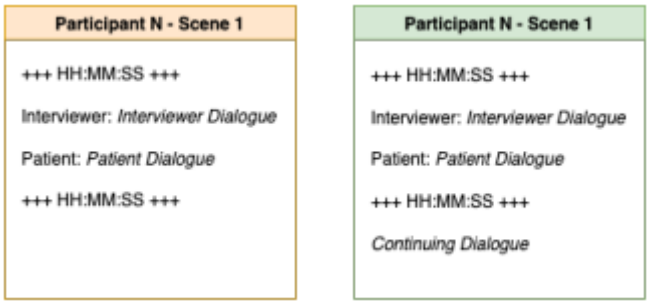

Figure 4: Transcription formats prior to preprocessing. The format at right was used when patient or interviewer utterances exceeded a given timestamp and continued onward into the next dialogue block.

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. In total we have 1100 conversation transcripts from 644 participants covering the three cases of Bipolar Disorder, Healthy Control, and Schizophrenia. Demographic details of the participants are presented in Table 1. The number of transcripts for each condition and SSPA scenario are presented in Table 2.

Category	Value
Mean Age	44.2
σ(Mean Age)	11.4
Females	58.4%
Males	41.3%
Unspecified	0.3%
African Americans	37.4%
American Indian/ Alaskan Native	0.5%
Asian	5.4%
White	48.3%
Multirace	7.0%
Hawaiian	0.6%
Unreported	0.6%

Table 1: Demographic details of the participants

Class * Scene	Number of Transcripts
BD Scene 1	225
BD Scene 2	223
HC Scene 1	103
HC Scene 2	103
SZ Scene 1	222
SZ Scene 2	225

Table 2: Number Transcripts by Scene and Condition

4.1.2 Model Details

In this section, we provide a brief description of models used in our project. We used 2 separate categories of models: encoder-decoder transformer models, and decoder only transformer models.

Encoder-Decoder Transformer Models We experiment with the T5-base(220M) and T5-large(770M) models. T5 [Raffel et al., 2020] is a transformer model based on the transformer architecture proposed in Vaswani et al. [2023]. T5 was developed after surveying the existing works on transfer learning in NLP so that the the good parts of every method is incorporated in T5. T5 formulates every NLP problem as text-to-text problem and is pre-trained on C4 that is a cleaned version of the Common Crawl Corpus.

Decoder-only transformer models We experiment with 3 separate decoder only transformer models, namely DialoGPT [Zhang et al., 2020b], TinyLlama-1.1B-Chat [Zhang et al., 2024], and Llama2-chat(7B) [Touvron et al., 2023]. For DialoGPT, we experiment with 2 different sizes of the model: DialoGPT-small, with 120M parameters and DialoGPT-large, with 770M parameters.

DialoGPT: DialoGPT [Zhang et al., 2020b] follows the same architecture as the GPT-2 [Radford et al., 2019] model. DialoGPT is trained on 147M Reddit conversation threads extracted from comment chains.

Llama2-chat: Llama2-chat[Touvron et al., 2023] is obtained by finetuning Llama2 model for dialogue use case. Llama2-chat is optimized for safety so that the chatbot refrains from responding with unhelpful or offensive content. Llama2-chat also introduces Ghost Attention that helps the chatbot focus on one personality during the entire conversation. To finetune Llama2 on dialogue data, the authors first start with instruction tuning data from Chung et al. [2022], before fine-tuning it on the annotated data collected by authors themselves. Llama2-chat is available in 7B, 13B, and 70B variants. We utilize the 7B variant for this project.

TinyLlama-1.1B-Chat: TinyLlama-1.1B-Chat follows the architecture of Llama-2[Touvron et al., 2023]. It is finetuned on a variant of UltraChat dataset [Ding et al., 2023], and UltraFeedback dataset [Cui et al., 2023].

4.1.3 Model Inputs:

Please refer to Section 3 for details on the content of input to the model. This section presents the way the input is fed to a transformer chat model. The chat models presented in the section above are trained with an input format specific to each model. Thus, utilizing the same input format during fine-tuning on our data is crucial for best model performance. To utilize the same input format, we use Chat Templates from HuggingFace-ChatTemplate. It converts a chat input to a format appropriate for each model. For instance, for TinyLlama1.1B-Chat model for the following chat:

System: You are a chatbot
User: How are you?
Assistant: I am good
the input fed to the model will be:
<|system|> You are a chatbot</s> <|user|> How are you?</s> <|assistant|> I am good</s> <|assistant|>

Here System message indicates a message at the beginning of the chat informing the model of the scenario and role it is expected to play.

4.1.4 Engineering Details

In this section, we present some of the engineering details that we employed to fine-tune larger transformer models such as Llama2-chat. We utilized the V100 GPUs with 34 GB VRAM. To reduce the training time of models, we employed parallel training via data parallelism. Data parallelism distributes the training batch across multiple GPUs such that each GPU receives the same batch size. We used the Accelerate library from HuggingFace-Accelerate to implement data parallelism. For data parallelism to work, the model should fit into a single GPU because the data is parallelized and not the model. The Llama2-chat

model, however, does not fit onto a single GPU *if all its parameters are to be finetuned.* Thus, to fit the Llama2-chat model onto a single GPU, we utilize Low Rank Adaptation [Hu et al., 2021] to fine-tune only a small percentage of total number of parameters.

Even after utilizing Low rank Adaptation(LoRA) to finetune only a subset of parameters, we still are unable to fit Llama2-chat onto a single GPU. To alleviate this, we utilize gradient checkpointing [Bulatov]. Generally, to calculate the gradients during the backwards pass, the activations from the forward pass are saved. Gradient checkpointing saves the space used to save these activations by only computing a subset of activations during forward pass so that only a subset of activations need to be re-computed during the backward pass. A downside of gradient checkpointing is that it makes the training slower since the activations need to be recomputed during the backward pass.

4.2 Evaluation Metrics

Evaluating open domain dialogue is challenging because there are a large number of possible appropriate responses. Most automatic evaluation metrics require a reference response be provided so that the model response can be judged against that. This can lead to false negatives as many appropriate responses are unfairly penalized. Thus, to evaluate the model performance, besides utilizing automatic evaluation metrics, we perform human evaluation as well. Section 4.2.1 presents the automatic evaluation metrics we used to evaluate the models. Section 4.2.2 presents the human evaluation metrics.

4.2.1 Automatic Evaluation Metrics:

To utilize automatic evaluation metrics, 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].

4.2.2 Human Evaluation:

We draw human evaluation metrics from [Tu et al., 2024], and from our domain knowledge of the task. We utilize 4 metrics for each of the two SSPA scenes described in Section 2.4. Out of the 4 metrics, three metrics are common across the scenes and 1 metric is specific to the scenes. The 3 common metrics are:

- **Stability/Consistency**: The role played by the chatbot should be the same over the entire dialogue. Moreover, the chatbot's response from a particular turn should not be contradictory to a response from a previous turn.
- **Coherency/Relevance**: The chatbot's response should be relevant to the user's utterance in the preceding turn and demonstrate a clear understanding of what the user said.
- **Fluency:** The response of the chatbot should be grammatically correct and free from any obvious errors.

The metrics specific to each task are:

- **Scene 1 specific metric:** The chatbot's responses should be brief. Also, the chatbot should not lead the user with any questions. The chatbot should just reciprocate and answer logically to the user's questions without taking conversational initiative. However, if the user seems to be stuck in the conversation, the chatbot can prompt them with a question.
- **Scene 2 specific metric:** The chatbot should not reach an easy compromise with the user. Rather, the chatbot should employ stalling tactics and try to give excuses to not fix the problem. The chatbot should not lead the user with any questions, but if the user seems to be stuck, the chatbot can prompt them with questions.

Each metric statement is present as a likert statement on a discrete scale of 1-4, with 1 being strongly disagree, and 4 being strongly agree. We got the models evaluated by two annotators.

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.20	0.31	0.18	0.20	0.31	0.18	0.79	0.72	0.75
BD Scene2	0.20	0.29	0.20	0.20	0.29	0.20	0.76	0.70	0.72
HC Scene1	0.22	0.33	0.20	0.22	0.33	0.020	0.81	0.73	0.76
HC Scene2	0.17	0.26	0.16	0.17	0.25	0.16	0.75	0.68	0.71
SZ Scene1	0.19	0.32	0.18	0.19	0.32	0.17	0.78	0.71	0.74
SZ Scene2	0.19	0.29	0.19	0.19	0.29	0.19	0.76	0.70	0.73

Table 3: DialoGPT-small results

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.26	0.37	0.25	0.25	0.36	0.25	0.80	0.74	0.77
BD Scene2	0.25	0.33	0.26	0.24	0.32	0.25	0.76	0.73	0.74
HC Scene1	0.29	0.38	0.28	0.28	0.38	0.28	0.81	0.76	0.78
HC Scene2	0.24	0.30	0.24	0.23	0.29	0.23	0.77	0.73	0.74
SZ Scene1	0.24	0.36	0.23	0.24	0.36	0.23	0.79	0.74	0.76
SZ Scene2	0.25	0.34	0.26	0.25	0.33	0.26	0.77	0.73	0.74

Table 4: DialoGPT-Large Results

4.3 Results

Table 3, Table 4, Table 5, Table 6, Table 7, and Table 8 give the results for DialoGPT-small, DialoGPT-large, TinyLlama-1.1B, T5-base, T5-large, and Llama2-chat respectively. Table 9 presents prelimiary results of human evaluation for TinyLlama-1.1B-chat model. Each cell represents average value of 20 transcript evaluations evaluated by two annotators. The observations from these results are presented in the next section.

4.4 Observations

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

1. From Table 5, we see that TinyLlama performs the best in terms of all the three metrics of BertScore, Rouge-1, and Rouge-L.

2. By comparing Table 3(DialoGPT-small) versus Table 4 (DialoGPT-large), and Table 6 (T5-base) versus Table 7 (T5-large), we see that the metrics measuring the syntax are better for larger model sizes, for a given architecture. This is true for BertScore metric as well. This indicates, that for a given architecture, the model is able to learn the task better with increase in model size.

3. Since Tiny Llama is the best performing model, it performs the best in both Scenes 1 and Scenes 2. However when TinyLlama, (Table 5) is compared against DialoGPT-large (Table 4) and T5-large (Table 3), we observe that the gap between TinyLlama and the two other models is larger for Scene 2 than Scene1 when compared on BertScore. This indicates that other models struggle to understand Scene 2 whereas TinyLlama does not.

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.32	0.36	0.35	0.31	0.35	0.34	0.79	0.79	0.79
BD Scene2	0.38	0.42	0.42	0.36	0.40	0.41	0.79	0.79	0.78
HC Scene1	0.31	0.36	0.35	0.31	0.35	0.34	0.80	0.79	0.79
HC Scene2	0.38	0.41	0.39	0.36	0.38	0.37	0.79	0.79	0.79
SZ Scene1	0.28	0.34	0.30	0.27	0.33	0.29	0.79	0.78	0.78
SZ Scene2	0.35	0.38	0.38	0.34	0.37	0.37	0.79	0.78	0.78

Table 5: 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	0.15	0.24	0.14	0.15	0.24	0.14	0.80	0.72	0.75
BD Scene2	0.14	0.25	0.13	0.14	0.25	0.13	0.77	0.67	0.71
HC Scene1	0.18	0.25	0.16	0.18	0.25	0.16	0.81	0.73	0.77
HC Scene2	0.12	0.21	0.11	0.12	0.20	0.11	0.75	0.67	0.71
SZ Scene1	0.15	0.27	0.14	0.15	0.27	0.14	0.79	0.71	0.74
SZ Scene2	0.12	0.22	0.12	0.12	0.22	0.11	0.76	0.67	0.71

Table 6: T5-base results

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.21	0.31	0.20	0.21	0.30	0.20	0.80	0.73	0.76
BD Scene2	0.21	0.33	0.20	0.21	0.33	0.20	0.79	0.70	0.74
HC Scene1	0.24	0.35	0.22	0.24	0.35	0.22	0.81	0.75	0.78
HC Scene2	0.19	0.28	0.18	0.19	0.28	0.18	0.77	0.70	0.73
SZ Scene1	0.20	0.34	0.19	0.20	0.33	0.19	0.79	0.72	0.75
SZ Scene2	0.16	0.26	0.16	0.16	0.26	0.16	0.78	0.69	0.73

Table 7: T5-large results

Class * Scene	Rouge1 F1	Rouge1 P	Rouge1 R	RougeL F1	RougeL P	RougeL R	BertScore P	BertScore R	BertScore F1
BD Scene1	0.12	0.10	0.35	0.11	0.09	0.32	0.64	0.75	0.69
BD Scene2	0.12	0.08	0.33	0.09	0.06	0.27	0.62	0.72	0.67
HC Scene1	0.12	0.09	0.35	0.11	0.09	0.33	0.65	0.76	0.70
HC Scene2	0.15	0.10	0.34	0.11	0.08	0.27	0.64	0.73	0.68
SZ Scene1	0.13	0.10	0.33	0.11	0.09	0.31	0.65	0.75	0.69
SZ Scene2	0.13	0.09	0.36	0.10	0.07	0.28	0.63	0.73	0.67

Table 8: Llama2-chat results

Class * Scene	Stability/Consistency	Coherency/Relevance	Fluency	Scene specific adherence
HC Scene 1	3.8	3.35	3.55	3.65
HC Scene 2	3.5	2.95	3.45	2.8
SZ Scene 1	3.55	3.3	3.45	3.3
SZ Scene 2	3.1	3.15	3.4	3.05
BD Scene 1	3.8	3.75	3.85	3.95
BD Scene 2	2.6	2.5	3.1	2.4

Table 9: Human Evaluation results for different scenes

4. From Table 8 we observe that the Llama2-chat model does not perform very well despite having the maximum no of parameters when compared to other models evaluated in this project. This can be attributed to the fact that we are unable to fine-tune all the parameters of Llama2-chat, and only fine-tune a small fraction of parameters. Furthermore, when the precision of BertScore is analyzed, it is consistently lower than recall across all the six scenarios. This indicates the model response does not contain a lot of the reference response. This is expected because we do not fine-tune all the parameters of Llama2-chat resulting in incomplete alignment with the dataset.

5 Future Work

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

1. As we observed that Llama2-chat, despite having more parameters than any other model, performs the worst. Whereas other models perform quite well according BertScore. This indicates that for our task, choosing an appropriate model size, that is closer to 1B patameters may be suitable.

2. It is common knowledge that LLMs are sensitive to the input prompts. Currently, we are feeding the models with prompts that we feel are appropriate for the given task. However, trying out prompt tuning techniques may improve the model performance further.

3. Our current method of making LLMs roleplay involves prompting the model with the scene information, and then fine-tuning on the data available. Trying out other available role-playing techniques is a promising next step.

4. Currently, the models are evaluated using automatic evaluation metrics Rouge and BertScore. Looking for other metrics than can more comprehensively evaluate the model performance is a good next step to take. Also, since the models are expected to be deployed with real patients, conducting thorough human evaluation beyond what has been done so far is crucial.

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 four different transformer based models: T5, DialoGPT, TinyLlama1.1B-Chat, and Llama2-Chat, and found that TinyLlama performed the best. Rigorously performing human evaluation of the models and trying out prompt-tuning techniques are promising next steps to extend this work.

References

Carlos Aguirre and Mark Dredze. Qualitative analysis of depression models by demographics. In Nazli Goharian, Philip Resnik, Andrew Yates, Molly Ireland, Kate Niederhoffer, and Rebecca Resnik, editors, *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 169–180, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.clpsych-1.19. URL https://aclanthology.org/2021.clpsych-1.19.

Silvio Amir, Mark Dredze, and John W. Ayers. Mental health surveillance over social media with digital cohorts. In Kate Niederhoffer, Kristy Hollingshead, Philip Resnik, Rebecca Resnik, and Kate Loveys, editors, *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*,

pages 114–120, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3013. URL https://aclanthology.org/W19-3013.

Yaroslav Bulatov. Gradient checkpointing. URL https://huggingface.co/docs/transformers/en/chat_templating.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Nicholas C. Coombs, Wyatt E. Meriwether, James Caringi, and Sophia R. Newcomer. Barriers to healthcare access among u.s. adults with mental health challenges: A population-based study. *SSM - Population Health*, 15:100847, 2021. ISSN 2352-8273. doi: https://doi.org/10.1016/j.ssmph.2021.100847. URL https://www.sciencedirect.com/science/article/pii/S2352827321001221.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

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. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature Communications*, 12(1):3242, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23458-5. URL https://doi.org/10.1038/s41467-021-23458-5.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *npj Digital Medicine*, 4(1):5, Jan 2021. ISSN 2398-6352. doi: 10.1038/s41746-020-00376-2. URL https://doi.org/10.1038/s41746-020-00376-2.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

HuggingFace-Accelerate. Accelerate documentation. URL https://huggingface.co/docs/accelerate/en/index.

HuggingFace-ChatTemplate. Chat template documentation. URL https://huggingface.co/docs/transformers/en/chat_templating.

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. Chatharuhi: Reviving anime character in reality via large language model, 2023.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Mpho Mokoatle, Vukosi Marivate, Darlington Mapiye, Riana Bornman, and Vanessa. M. Hayes. A review and comparative study of cancer detection using machine learning: Sbert and simcse application. *BMC Bioinformatics*, 24(1):112, Mar 2023. ISSN 1471-2105. doi: 10.1186/s12859-023-05235-x. URL https://doi.org/10.1186/s12859-023-05235-x.

Thomas L Patterson, Sherry Moscona, Christine L McKibbin, Kevin Davidson, and Dilip V Jeste. Social skills performance assessment among older patients with schizophrenia. *Schizophrenia Research*, 48 (2):351–360, 2001. ISSN 0920-9964. doi: https://doi.org/10.1016/S0920-9964(00)00109-2. URL https://www.sciencedirect.com/science/article/pii/S0920996400001092.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role-play with large language models, 2023.

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

Eli Sherman, Keith Harrigian, Carlos Aguirre, and Mark Dredze. Towards understanding the role of gender in deploying social media-based mental health surveillance models. In Nazli Goharian, Philip Resnik, Andrew Yates, Molly Ireland, Kate Niederhoffer, and Rebecca Resnik, editors, *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 217–223, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.clpsych-1.23. URL https://aclanthology.org/2021.clpsych-1.23.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Quan Tu, Shilong Fan, Zihang Tian, and Rui Yan. Charactereval: A chinese benchmark for role-playing conversational agent evaluation, 2024. URL https://arxiv.org/abs/2401.01275.

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

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. Tinyllama: An open-source small language model, 2024.

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

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