AI-Driven Mental Health Support Assistant
Deep Learning for Data Science
DD2424

Abdul Fathaah Shamsuddin
Prashant Yadava

May 27, 2024

KTH Royal Institute of Technology

Abstract

Building large language models(LLMs) as a psychologist in your pocket,

This project investigates the feasibility of training a small Large Language Model (LLM) to provide high-quality psychological advice. We begin by training a Recurrent Neural Network (RNN) on a dataset of psychological advice, but find its performance to be underwhelming. We then fine-tune a pre-trained LLM, Llama-3-8b, on the samhog/psychology-10k dataset, achieving promising results. To further improve the model's performance, we attempt to generate synthetic training data using GPT-3.5 Turbo, a powerful language generation model. Additionally, we explore methods to refine the responses of the LLM to generate better training data, creating a self-improving cycle. Our findings suggest that a specialized LLM can be trained to provide effective psychological advice, and that generating high-quality synthetic data could improve this process significantly. This project contributes to the development of AI-powered mental health support systems and highlights the potential of LLMs in this domain.

Keywords: LLM · Fine-tuning · LoRA · Llama3

Contents

List of Tables

List of Figures

Chapter 1

Theory

1.1 Text Generation

Text generation is an important task in the natural language processing domain. With the recent rise in LLMs[15] it has become easier to generate text for specific use cases. But unsolved challenges like hallucination of LLMs, accuracy, domain specific knowledge and more still remains.

Earliest indication of text generation capabilities were first demonstrated using RNNs, LSTMS [11].

1.2 RNN & LSTM

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been pivotal in the evolution of text generation techniques in Natural Language Processing (NLP).

1.2.1 Recurrent Neural Networks (RNNs)

RNNs are designed to handle sequential data by maintaining a hidden state that captures information from previous time steps. This makes them suitable for tasks like text generation, where the context of previous words influences the generation of subsequent words. However, standard RNNs suffer from the vanishing gradient problem, where gradients diminish exponentially as they are backpropagated through time, making it difficult to learn long-term dependencies[9].

1.2.2 Long Short-Term Memory (LSTM)

LSTM networks were introduced to address the limitations of standard RNNs. LSTMs incorporate memory cells and gating mechanisms (input, forget, and output gates) that regulate the flow of information, allowing them to maintain long-term dependencies more effectively. This architecture prevents the vanishing gradient problem, enabling LSTMs to learn and generate coherent text over longer sequences[9]. LSTMs have been widely used in text generation tasks, outperforming traditional RNNs in generating more contextually relevant and grammatically correct text.

1.2.3 Evolution to Transformers

Despite the advancements with LSTMs, their sequential nature limits parallelization, making training time-consuming. Transformers, introduced in 2017, revolutionized NLP by using self-attention mechanisms that allow for parallel processing of data. This architecture enables transformers to capture dependencies across entire sequences without the need for sequential processing, significantly improving efficiency and performance in text generation tasks[1]. Transformer-based models, such as GPT and BERT, have set new benchmarks in generating human-like text, offering superior fluency and coherence compared to RNNs and LSTMs[5].

In summary, while RNNs laid the groundwork for sequential data processing, LSTMs improved upon them by addressing long-term dependency issues. The advent of transformers has further advanced the field, providing more efficient and powerful tools for text generation.

1.3 Large Language Models

Large language models (LLMs) are a type of artificial intelligence model that uses deep learning techniques and massive datasets to understand, summarize, generate, and predict text-based content with human-like capabilities [15]. LLMs like GPT-3, BERT, and others are typically pre-trained on vast amounts of general text data from the internet. This pre-training allows them to acquire broad language understanding capabilities.

While pre-training provides a strong foundation, these general-purpose LLMs may not perform optimally on specialized tasks or domains without further adaptation. To address this, a process called fine-tuning [1] is employed. Fine-tuning a large language model refers to adapting a pre-trained LLM to a specific task, domain, or dataset by further training it on relevant examples . During fine-tuning, the model learns patterns and knowledge relevant to the target task, enabling it to generate more accurate and contextually relevant outputs for that domain.

1.4 Fine Tuning LLMs

The fine-tuning process involves taking the pre-trained LLM weights as a starting point and further training (updating) the model's parameters on a smaller, task-specific dataset . This dataset typically consists of input-output pairs that demonstrate the desired behavior for the target task, such as question-answer pairs, text-summary pairs, or code-documentation pairs. Fine-tuning allows enterprises to customize general LLMs to their proprietary data, domain knowledge, and specific use cases, without the massive computational resources required for training from scratch.

1.4.1 LoRA Low Rank Adapatation

LoRA (Low-Rank Adaptation) is a technique for efficiently fine-tuning large pre-trained models on new datasets or tasks while significantly reducing computational requirements and memory footprint [16]. Instead of updating all parameters during fine-tuning, LoRA introduces a small number of trainable parameters called "update matrices" or "LoRA weights". These low-rank matrices have a reduced number of parameters compared to the full weight matrices of the pre-trained model.

During fine-tuning, only the LoRA weights are trained and updated [12], while the original pre-trained weights remain frozen. The LoRA weights are then combined with the frozen pre-trained weights through a mathematical operation called "low-rank decomposition" to produce the final weight matrices used for inference. This approach leverages the "intrinsic rank hypothesis", which suggests that significant changes to a neural network can be captured using a lower-dimensional representation, i.e., the low-rank LoRA weights.

The key benefits of LoRA include reduced memory footprint, faster training and adaptation, smaller model size, and preservation of pre-trained knowledge. By only training a small number of LoRA weights, the memory requirements are significantly lower, enabling fine-tuning on consumer-grade GPUs or even CPUs. With fewer parameters to update, the training process is accelerated, allowing for quicker adaptation to new tasks or domains. Additionally, the trained LoRA weights are much smaller than the full model, making it easier to store and share the fine-tuned models. Since the original weights remain frozen, the pre-trained model's knowledge is largely preserved, reducing the risk of catastrophic forgetting.

1.4.1.1 Use of LoRA for fine-tuning in the Project

LLMs, for instance, Meta's Llama models [6], although excelling at capturing language nuances, contextual understanding, and complex tasks like translation and dialogue generation, generally do not perform well on tasks which require domain expertise, such as mental health or psychiatry related topics.

One of the high-level goals of our project is to take a large language model (LLM) and fine tune it with a psychiatry dataset so that the model could be used to get answers to psychiatry/ mental health related questions.

For the purpose of the project, we decided to experiment with the recently released Llama 3 models [5]. On 18th March 2024, Meta released 2 versions of Llama 3 model, one with 8 billion parameters, and the other with 70 billion parameters.

	Meta Llama 3 8B	Gemma 7B - It Measured	Mistral 7B Instruct Measured
MMLU 5-shot	**68.4**	53.3	58.4
GPQA 0-shot	**34.2**	21.4	26.3
HumanEval 0-shot	**62.2**	30.5	36.6
GSM-8K 8-shot, CoT	**79.6**	30.6	39.9
MATH 4-shot, CoT	**30.0**	12.2	11.0

	Meta Llama 3 70B	Gemini Pro 1.5 Published	Claude 3 Sonnet Published
MMLU 5-shot	**82.0**	81.9	79.0
GPQA 0-shot	39.5	**41.5** CoT	38.5 CoT
HumanEval 0-shot	**81.7**	71.9	73.0
GSM-8K 8-shot, CoT	**93.0**	91.7 11-shot	92.3 0-shot
MATH 4-shot, CoT	50.4	**58.5** Minerva prompt	40.5

Fig. 1.1: Meta Llama 3 Instruct model performance

1.5 Synthetic Data Generation

As mentioned in the previous sections, fine-tuning with a domain specific data can enhance the capabilities of a pre-trained LLM for that particular domain.

However, obtaining high-quality, task-specific datasets for fine-tuning can be challenging, time-consuming, and expensive, especially for niche domains or applications like psychiatry or psychology. This is where synthetic data generation has emerged as a promising approach.

Synthetic data generation involves using the pre-trained LLM itself or other language models to generate synthetic examples that mimic the desired task or domain. This approach has several advantages, including scalability, customization, privacy preservation, and cost-effectiveness.

By generating synthetic data, LLMs can be fine-tuned on large quantities of tailored data, enabling more targeted adaptation to specific tasks or domains. This approach avoids the need for manually annotating or collecting real-world data, mitigating privacy and copyright concerns. Additionally, synthetic data generation is generally more cost-effective than traditional data collection methods. However, it's important to note that synthetic data generation also has limitations, such as the potential for introducing biases or artifacts from the generating model, and the need for careful curation and validation of the generated data.

We utilized synthetic data generation for multiple goals:

- generation of dataset for training and refining our model, and
- improving the quality of response provided by the psychologist.

1.5.1 Generation of training data

We utilized Langchain [3] and OpenAI's ChatGPT gpt-3.5-turbo [2] for data generation for domain specific psychiatry help dataset. We used the Llama 3 70 billion parameter model for this task, where we generated conversations between a psychiatrist and a patient seeking help with mental health support.

Langhcain provides a python library [4] that enables building applications with LLMs while providing support for various use cases, including synthetic data generation. It is designed to make it easier to build applications that leverage the capabilities of LLMs, such as text generation, question answering, and data analysis. It provides a modular and extensible architecture that allows developers to combine different components, such as LLMs, prompts, and data sources, to create custom workflows and applications.

1.5.2 Refining the Synthetic Data using LLM Agent

We used OpenAI's `gpt4 turbo` APIs and the Microsoft's Autogen [7] to create `Assistants` and obtained refined psychologist response.

Chapter 2

Models and Test Methodology

This Chapter goes into the details of the models that were created for the project and the test methodology. The implementation for the project is available at the github repository AI-Driven Psychologist [17].

2.1 LLM Model Training Background

2.1.1 Model variants

We have chosen 3 variations of the model, Llama-3 8B for the comparison of generated output. These 3 variants are:

- Llama-3 8B model without fine-tuning
- Llama-3 8B model with fine-tuning, without gradient accumulation
- Llamma-3 8B model with fine-tuning, with gradient accumulation

Model Description	Additional Notes
Llama-3 8B model without fine-tuning	Fine-tuning with LoRA was not attempted, and hence this model is not enriched with psychology domain knowledge.
Llama-3 8B model with fine-tuning, without gradient accumulation	Fine-tuning with LoRA was done, but without a gradient accumulation.
Llama-3 8B model with fine-tuning, with gradient accumulation	Fine-tuning with LoRA was done, and a 4-step gradient accumulation was done. the weights of the models are updated with a gradient update for batch of data instead of a single data point, leading to potentially better convergence and more stable training.

Table 2.1: Model variations and their descriptions

The prompt to the model was set up to include 3 parts:

- **Instruction**: The instruction to the LLM model to assume the role of a psychologist and give helpful response to the user's input.
- **Input**: the user question asked to the LLM.
- **Output**: the response received from the LLM.

2.1.2 Training Data for fine-tuning

For fine-tuning the model, i.e., bringing in the domain expertise from psychology into the models, we use Huggingface dataset samhog/psychology-10k [8], which is a collection of 10,000 psychology-related prompts and responses generated by ChatGPT. This dataset was created by Samuel Höglund as part of a thesis project in machine learning and psychology at KTH Royal Institute of Technology. It contains 10,000 question-answer pairs related to various topics in psychology, such as mental health, human behavior, and psychological theories.

To speed up fine-tuning, we used Unlsoth [13], which is an open-source toolchain designed for efficient fine-tuning of large language models (LLMs) like Llama, Mistral, and Gemma on GPUs. It accelerates the fine-tuning process by manually optimizing compute-heavy mathematical operations and handwriting GPU kernels. This allows for faster training without requiring hardware changes [14].

2.1.3 Fine tuning models using LoRA

Fine tuning of the models using LoRA was done using the following approach:

Pseudocode 1 High-Level Overview of LoRA Approach

1: **Step 1:** Load pre-trained model weights.
2: **Step 2:** Initialize LoRA weights (low-rank matrices).
3: **Step 3:** Fine-tune the model:

- **a.** Perform forward pass using the training data.
- **b.** Compute the loss based on the model outputs and targets.
- **c.** Compute gradients from the loss.
- **d.** Update only the LoRA weights using the computed gradients.

4: **Step 4:** Combine the pre-trained weights with the updated LoRA weights.
5: **Step 5:** Use the combined weights for inference.

This helped in getting the domain expertise from the psychology dataset `samhog/psychology-10k` obtained from Huggingface into the models.

2.2 Comparison of results from the Models

2.2.1 Qualitative Comparison Criteria

We picked the 3 questions from the training data to pass as the input to the LLM models. The output response from the models was then compared qualitatively (human evaluation) with the answer in the training dataset.

Input to the LLM	Output Response from the training data
"I can't stop thinking about my ex-boyfriend. What should I do?"	"It's normal to have thoughts about an ex, but if it's interfering with your daily life, we can work on strategies to help you move on. This can include practicing mindfulness techniques, focusing on self-care and self-love, and exploring any underlying issues that may be contributing to your attachment."
"I'm struggling with anxiety and it's affecting my daily life. What can I do?"	"It's great that you're seeking help. We can work together to identify your triggers and develop coping strategies such as deep breathing exercises or mindfulness techniques."
"I'm feeling really overwhelmed with work and family responsibilities. I don't know how to manage everything."	"It's important to prioritize your responsibilities and delegate tasks when necessary. Let's work on developing time management skills and finding ways to reduce your stress levels."

Table 2.2: Table showing 3 pairs of input-response from the training data.

2.2.2 Quantitative Comparison Criteria

For a quantitative comparison of the models' response, we used cosine similarities between the output from the LLM with the response for the same input in the training dataset.

The `cosine_similarity` function computes the cosine similarity between two embeddings. Cosine similarity measures the cosine of the angle between two non-zero vectors, which is useful for determining the similarity between different data points, such as word or sentence embeddings.

Function Definition

```
import torch.nn.functional as F
def cosine_similarity(embedding1, embedding2):
    score = 0
    for i in range(embedding1.shape[1]):
      for j in range(embedding2.shape[1]):
        score += F.cosine_similarity(embedding1[:,i,:], embedding2[:,j,:])
            .item()
    return score / (embedding1.shape[1] * embedding2.shape[1])
```

Listing 2.1: Cosine Similarity Function

Description of `cosine_similarity` Function

The `cosine_similarity` function computes the cosine similarity between two embeddings. Cosine similarity measures the cosine of the angle between two non-zero vectors, which is useful for determining the similarity between different data points, such as word or sentence embeddings.
Parameters:

- **embedding1**: The first embedding tensor, typically a multi-dimensional array where each vector represents an embedding.
- **embedding2**: The second embedding tensor, similar in structure to `embedding1`.

Returns: A scalar value representing the average cosine similarity score between the embeddings.
Function Steps:

1. Initialize the score variable to 0. This accumulates the sum of cosine similarity scores between all pairs of vectors from `embedding1` and `embedding2`.
2. Use two nested `for` loops to iterate over the second dimension (column-wise) of `embedding1` and `embedding2`.
   - The outer loop iterates over the columns of `embedding1`.
   - The inner loop iterates over the columns of `embedding2`.
3. For each pair of vectors `(embedding1[:,i,:], embedding2[:,j,:])`, compute the cosine similarity using `F.cosine_similarity` and add the result to the score.
4. Normalize the score by dividing it by the product of the number of columns in `embedding1` and `embedding2` to obtain the average cosine similarity.

2.3 Synthetic Data Generation

we used the Langchain [3] framework and OpenAI's ChatGPT LLM Model `gpt-3.5 turbo` [2] for generating the synthetic data containing psychologist and patient conversations.

Using Langchain, we created agents representing both the patient and the psychiatrist. The psychiatrist agent was instructed to act like a licensed psychologist and respond to the patient's mental health support request.

2.4 Refining the Synthetic Data using LLM Agent

We also used OpenAI's gpt3.5 turbo APIs to create a refinement of the initial response given by the psychologist to a patient. Here, we created a new psychologist, a so-called

Expert Evaluator Psychologist who was tasked to analyze the original response from the psychologist and provide it feedback on how to improve its response. As a result we had an improved higher quality response for the patient.

Function Definition

```
name="Psychologist",
instructions="If you are a licensed psychologist, please provide this
    patient with a helpful response to their concern.."
```

Listing 2.2: Instructions for Psychologist Assistant

```
name="ExpertPsychologistEvaluator",
instructions="You are an expert psychologist with extensive knowledge
    and experience in mental health. Your role is to evaluate the
    responses provided by a psychologist to a patient seeking help for
    mental health issues. When a psychologist's response is provided to
     you, you should critically analyze it and provide constructive
    feedback. Consider the following aspects: - Accuracy and
    correctness of the information provided, - Empathy and sensitivity
    towards the patient's concerns, - Clarity and understandability of
    the language used, Appropriateness and relevance of the advice or
    suggestions given, Any potential gaps or missing information that
    should be addressed. Your feedback should be professional,
    objective, and aimed at helping the psychologist improve their
    responses. Provide specific suggestions or recommendations for
    improvement, but avoid being overly critical or dismissive.
    Remember, your goal is to ensure that the patient receives high-
    quality, effective, and compassionate support from the psychologist
    ."
```

Listing 2.3: Instruction for the Expert Evaluator Psychologist

CHAPTER **3**

Results

3.1 Results of finetuning of LLM

Multiple rounds of finetuning was done on the samhog/psychology-10k[8] data with different parrameter. Table 3.1 shoes how the results look in different settings. r is the rank of the LoRa model, it determines how many parameters are present for the model. When $r = 16$, we have 41 million parameters and $r = 32$ it is 82 million parameters for llama-3-8b. These results were obtained on a 4-bit quantized model which made it possible to run it on a Google Colab T4 GPUs with 16GB RAM.

In traditional backpropagation, the gradients of the loss function with respect to the model's parameters are computed and updated after each mini-batch. However, when using gradient accumulation, the gradients are accumulated (i.e., summed) over multiple mini-batches before updating the model's parameters. It's particularly useful when working with limited GPU memory or when dealing with large models and batch sizes. Table 3.1 represents results with gradient accumulation of 4 steps, it is worth noting that without gradient accumulation the model tends to overfirt on individual training samples than a batch. Hence, individual text comparisons achieve higher without gradient accumulation.

Model	**Passage Ranking**	**Semantic Textual Similarity**	**Llama-8b**
Trained 300 (without grad accumalation), $r = 16$	1.0000	0.9999	0.3181
Trained 300 epochs, $r = 16$	0.9749	0.9999	0.3145
Trained 300 epochs, $r = 32$	0.7104	0.8207	0.0918
Trained 900 epochs, $r = 32$	0.7094	0.9279	0.3136
No finetuning	0.7105	0.9286	0.3137

Table 3.1: Model Performance Results

We use passage ranking and semantic textual similarity[10] for comparing the expected output and the output from the models. We also leveraged the methodology described in section 2.2.2 to evaluate as well, which utilised the embeddings of llama-3-8b model. In reality evaluating these responses be subjective and difficult to quantify. In the scope of this project we utilise 50 data points from the dataset[8] to test the model. We can notice how the models are able to fit well on the data when $r = 16$. It cannot be concluded that for $r = 32$, the model performs pooly, due to lack of time we were not able to tune the hyperparameters for optimal results.

3.2 Poor results from RNN & LSTM

Although we attempted to train Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks on the data, these models struggled to fit the data and yielded poor results. Our minimal experimentation with these models revealed that the dataset was particularly challenging to model, primarily due to its requirement for holding long contextual information. The complexity of the data made it difficult for these models to capture the underlying patterns and relationships, ultimately leading to subpar performance.

Output from RNN

Patient: I can't stop thinking about my ex-boyfriend . What should I do ?
RNN response: relationship persist lose lose lose lose lose lose lose lose lose lose lose lose lose lose pills prayer electronics

3.3 Synthetic Data Generation

As mentioned earlier, we generated the synthetic data containing psychologist and patient conversations using the LLM model `gpt-3.5 turbo` and `Langchain` framework. We generated 984 samples to train the model and achieve results discussed in chapter 3.4.

Below is a sample of the generated dataset:

Message from the Patient to the Psychologist

I'm feeling really anxious lately and I don't know why.

Response from Psychologist to Patient

As a licensed psychologist, I want to reassure you that feeling anxious is a common experience that many individuals face. It's important to explore the root cause of your anxiety, whether it be triggered by past experiences or current stressors. I recommend practicing mindfulness techniques such as deep breathing exercises, meditation, or yoga to help calm your mind and body. Additionally, seeking professional help through therapy can provide you with the necessary tools to manage your anxiety effectively. Remember, you are not alone in this journey, and there is always support available to guide you towards a place of inner peace and emotional well-being.

3.4 Training Loss

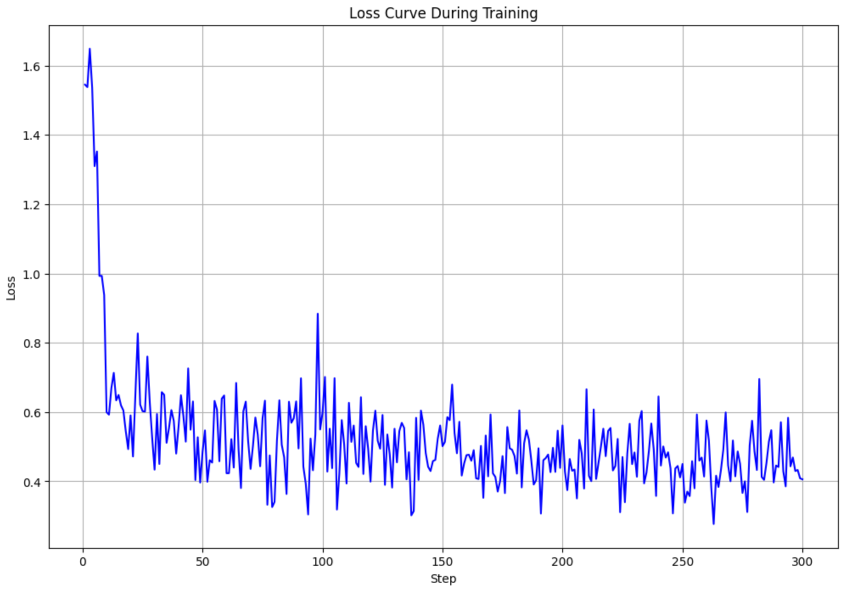

Fig. 3.1: Training loss curve for fine-tuning Llama 3 8B model using LoRA with rank 16

From our experiments, the fine-tuned model `Llama 3 8B` using LoRA with rank 16 matrices had the best performance in terms of the quality of results obtained. Figure 3.1 shows the progression of loss during training of the model. The loss curve suggests that the model is learning effectively, especially in the early stages of training. In the middle part of the training the learning rate seems to have slowed down, and improvements are happening more gradually.

In this context it is worth mentioning that the generated data (response from the Psychologist) in the training dataset usually had the text `"As a licensed psychologist, ..."`. This might have played a role in the loss dropping significantly in the beginning part of the training as the model "learned" the pattern quickly.

3.5 Generated Text Results

In the experiments, the LLM was instructed to be a “licensed psychologist” and provide a “helpful response” to a patient.

Below is an example how the prompt looked like:

Prompt structure

"alpaca_prompt.format("If you are a licensed psychologist, please provide this patient with a helpful response to their concern..", # instruction "I'm feeling really overwhelmed with work and family responsibilities. I don't know how to manage everything.", # input "", # output - leave this blank for generation!)"

It was interesting to note that even though instructed to be a licensed psychologist, the LLM was unable to fulfil the requirement. Below is the response from the LLM, which clearly doesn't sound as if it is from a 'licensed psychologist', a role which it was instructed to assume. It, instead, advises the user to seek help from a licensed professional.

Response from Psychologist to Patient

It sounds like you are feeling a lot of stress. Have you considered seeking help from a professional? A therapist or counselor can help you manage your stress and develop healthy coping mechanisms. Additionally, it's important to take care of yourself physically and mentally. Make sure you are getting enough sleep, eating a balanced diet, and engaging in activities that you enjoy. Remember that you are not alone in this and that there are resources available to help you.

3.5.1 Synthetic data generation for Response refinement

Below we present our use of synthetic data generation method to obtain refined response from psychologist. This is a promising approach and get be extended in many ways. We can ask the expert psychologist evaluator to score the responses. This score can be used further for training in a reinforcement led way to train the model.

Initial Message from Patient to Psychologist
I'm feeling really overwhelmed with work and family responsibilities. I don't know how to manage everything.

Initial Response from Psychologist to Patient
Thank you for sharing your feelings of being overwhelmed with work and family responsibilities. It's completely normal to feel this way when trying to juggle multiple obligations. It's important to take steps to manage your stress and prioritize your well-being. Here are some tips that may help you navigate this challenging situation:
1. Start by acknowledging your feelings and giving yourself permission to feel overwhelmed. Remember that it's okay to ask for help and take breaks when needed.
2. Prioritize tasks by creating a to-do list and focusing on one task at a time. Break down larger tasks into smaller, more manageable steps to prevent feeling overwhelmed.
3. Consider delegating tasks to family members or colleagues to lighten your load. Don't be afraid to ask for support when you need it.
4. Set boundaries to protect your time and energy. Learn to say no to additional commitments that may add to your stress.
5. Practice self-care by prioritizing activities that help you relax and recharge. This could include exercise, mindfulness techniques, or engaging in hobbies you enjoy.
6. Consider seeking professional support from a counselor or therapist to explore coping strategies and develop a personalized plan for managing your stress and responsibilities.
Remember, it's important to prioritize your well-being and seek support when needed. You don't have to navigate this alone. Take small steps towards managing your responsibilities and remember to take care of yourself along the way.

Instruction for refining the response from Expert Psychologist Evaluator to Psychologist

Your response contains some helpful suggestions for managing overwhelming work and family responsibilities. Here are some points for improvement:
1. Empathy and Validation: It's crucial to acknowledge the patient's feelings of being overwhelmed with empathy and validation before jumping into solutions. Starting with a statement like, "I'm sorry to hear that you're feeling overwhelmed. It's completely understandable given your current situation," can help the patient feel heard and understood.
2. Individualized Advice: While your suggestions are generally useful, it would be beneficial to tailor the advice more specifically to the patient's situation. For example, understanding the specific aspects of work and family responsibilities causing the most stress could help in providing more targeted advice.
3. Additional Coping Strategies: In addition to delegating tasks, setting boundaries, and practicing self-care, it could be helpful to suggest other coping strategies such as mindfulness exercises, time management techniques, or seeking support groups for individuals in a similar situation.
4. Referral to Professional Help: Mentioning the option of seeking professional help earlier in the response could be beneficial. Encouraging the patient to consider therapy or counseling as a valid form of support for managing overwhelm can be empowering.
Overall, your response provides a good starting point for managing overwhelming responsibilities, but incorporating more empathy, individualization, additional coping strategies, and early mention of professional help could enhance the support provided to the patient.

Refined response from Psychologist to Patient

I understand that feeling overwhelmed by work and family responsibilities can be challenging. It's important to prioritize tasks and set boundaries to maintain a healthy balance. Here are some tips that may help you manage your workload and reduce stress:
1. Prioritize tasks: Make a list of your tasks and prioritize them based on urgency and importance. Focus on completing high-priority tasks first and tackle the rest later.
2. Delegate responsibilities: Don't be afraid to ask for help. Delegate tasks to family members or colleagues when possible to lighten your load.
3. Set boundaries: Establish boundaries to protect your time and energy. Learn to say no to additional commitments that may lead to overwhelm.
4. Practice self-care: Take care of your physical and mental well-being by getting enough rest, eating healthily, exercising, and finding time for activities you enjoy.
5. Seek support: Talk to your family or friends about how you're feeling. Consider seeking professional support from a counselor or therapist if needed.
Remember, it's okay to ask for help and prioritize your well-being. You don't have to do it all alone. Take small steps towards managing your workload and take care of yourself.

Chapter 4

Conclusion

In conclusion, our project demonstrates the promising potential of Large Language Models (LLMs) in providing effective psychological advice. The ease with which our model was fine-tuned on the data and the promising results obtained suggest that LLMs can be a valuable tool in this domain. Furthermore, we propose two innovative approaches to generating high-quality training data: synthetic data generation and synthetic data generation in a group chat format with response refinement. These techniques have the potential to significantly enhance the model's performance and capabilities. Our projects also highlights the limitations of traditional Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks in modeling long contextual information, underscoring the need for more advanced architectures like LLMs. Overall, our findings have important implications for the development of AI-powered mental health support systems and suggest a promising future for LLMs in this field.

Bibliography

[1] Fine-tuning large language models, https://learn.microsoft.com/en-us/ai/playbook/technology-guidance/generative-ai/working-with-llms/fine-tuning, accessed: 2024-05-22

[2] Gpt-3.5 turbo documentation, https://platform.openai.com/docs/models/gpt-3-5-turbo, accessed: 2024-05-24

[3] Langchain, https://www.langchain.com/, accessed: 2024-05-24

[4] Langchain documentation, https://python.langchain.com/v0.1/, accessed: 2024-05-23

[5] Llama 3: Open and efficient foundation language models, https://llama.meta.com/llama3/, accessed: 2024-05-22

[6] Llama: Open and efficient foundation language models, https://llama.meta.com/, accessed: 2024-05-22

[7] Microsoft autogen, https://microsoft.github.io/autogen/, accessed: 2024-05-27

[8] Psychology 10k dataset, https://huggingface.co/datasets/samhog/psychology-10k/viewer, accessed: 2024-05-22

[9] Recurrent neural network, https://en.wikipedia.org/wiki/Recurrent_neural_network, accessed: 2024-05-23

[10] Semantic similarity results, https://huggingface.co/tasks/sentence-similarity, accessed: 2024-05-24

[11] Text generation with rnn, https://www.tensorflow.org/text/tutorials/text_generation, accessed: 2024-05-23

[12] Training with lora (low-rank adaptation), https://huggingface.co/docs/diffusers/en/training/lora, accessed: 2024-05-22

[13] Unsloth ai, https://unsloth.ai/, accessed: 2024-05-23

[14] Unsloth ai github repository, https://github.com/unslothai/unsloth, accessed: 2024-05-23

[15] What is a large language model (llm)?, https://www.techtarget.com/whatis/definition/large-language-model-LLM, accessed: 2024-05-22

[16] Miller, T.: Explainable ai: A systematic review and comparative analysis (2024), https://arxiv.org/pdf/2404.10779, accessed: 2024-05-22

[17] smartwhale8: Ai-driven psychologist (2024), https://github.com/smartwhale8/ai-driven-pscychologist, accessed: 2024-05-24