Naptick AI Challenge - Task 2: Voice-to-Voice Sleep Coaching Agent

Submitted by: Jayant Aggarwal
Position Applied For: Intern
Date: 24/05/2025

1. Introduction

This document details the implementation of a voice-to-voice intelligent sleep coaching agent, developed for Task 2 of the Naptick AI Challenge. The primary goal was to create an AI assistant capable of understanding user queries about sleep (via voice or text) and providing specialized, helpful, and safe advice. A key component of this task was to fine-tune a Large Language Model (LLM) using custom sleep-domain data to enhance the relevance and quality of its responses, including handling multi-turn conversations with evolving advice.

2. System Architecture & Models Used

The agent operates through the following pipeline:

1. **User Input:** Accepted as text or voice (live recording/audio file upload) via a Gradio web interface.
2. **Speech-to-Text (STT):** Voice input is transcribed using faster-whisper (specifically, the base.en model), chosen for its balance of accuracy and efficiency on a Colab T4 GPU environment (using float16 compute).
3. **LLM Processing:** The transcribed text (or direct text input) is processed by a fine-tuned mistralai/Mistral-7B-Instruct-v0.2 model.
    - **Base Model Loading:** The model was loaded using 4-bit quantization via the bitsandbytes library to manage memory constraints in Colab.
    - **Fine-Tuning:** Parameter-Efficient Fine-Tuning (PEFT) with LoRA (Low-Rank Adaptation) was employed using Hugging Face transformers, peft, and datasets libraries.
    - **Conversation History:** The LLM maintains context through a conversation history list, enabling multi-turn dialogue.
4. **Text-to-Speech (TTS):** The LLM's textual response is synthesized into audible speech using piper-tts with the en_US-lessac-medium voice model. TTS pre-processing (e.g., converting "7-9" to "7 to 9") was implemented for more natural pronunciation.
5. **Output:** The agent provides both the text transcript of its response and an auto-playing audio version via the Gradio interface.

All development and execution were performed within a Google Colab notebook environment.

3. Fine-tuning Data

A custom dataset was curated for fine-tuning the Mistral-7B model, consisting of approximately **[90+]** instruction-response pairs in .jsonl format. This dataset was designed to cover the specific requirements of the challenge:

- **Simulated Wearable Data Samples:** Examples included interpreting summarized data from hypothetical wearable devices (e.g., TST, efficiency, sleep stages, RHR, HRV from "Whoop," "Fitbit," etc.) and answering user questions based on this data.
    - *Example:* {"instruction": "My Whoop report: 7h 10m Time in Bed, 6h 35m Sleep, 89% Efficiency. Stages: 15% Awake, 25% REM, 45% Light, 15% Deep. RHR 55bpm, HRV 68ms. Felt pretty good waking up. Is this a good sleep profile?", "response": "Based on your report, that looks like a solid night! ..."}
- **Sleep Diary Interpretations:** Scenarios combining subjective diary entries (e.g., bedtime, feelings, disruptions) with objective (simulated) sleep stage data to provide contextual advice.
    - *Example:* {"instruction": "Sleep Diary: Bed 10:30 PM... Mind felt busy... Woke up ~4 AM... Feel wrecked. Had a stressful work day.", "response": "That sounds like a really rough night... High stress from your workday very likely contributed..."}
- **Research-Backed Sleep Knowledge:** Q&A pairs covering sleep science concepts (stages, cycles, hygiene), biological mechanisms (hormones like melatonin, orexin, ghrelin/leptin; processes like the glymphatic system), and the effects of various factors (light, temperature, caffeine, stress, exercise, diet) on sleep.
    - *Example:* {"instruction": "What's the connection between stress and sleep?", "response": "Stress and sleep have a strong, two-way relationship. High stress levels activate the body's 'fight-or-flight' response..."}
- **Multi-Turn Conversation & Evolving Advice (Bonus):** Examples were included where the user refers back to previous advice or provides new context, requiring the agent to adapt and offer different or more refined suggestions.
    - *Example (Conceptual):* {"instruction": "User: I tried avoiding screens, but my mind still races. What else?", "response": "Okay, good you tried limiting screen time! Since your mind is still racing, let's focus on mental relaxation techniques like meditation or deep breathing..."}

The dataset was carefully crafted to ensure responses were informative, safe (e.g., deferring medical questions to professionals), and aligned with a supportive coaching persona.

(You can include 2-3 more diverse examples from your dataset here)

4. Fine-tuning Process

The fine-tuning was performed in Google Colab using the following PEFT/LoRA setup:

- **Base Model:** mistralai/Mistral-7B-Instruct-v0.2 (loaded in 4-bit)
- **Libraries:** transformers.Trainer, peft.LoraConfig, peft.get_peft_model
- **Key Hyperparameters:**
    - num_train_epochs: 3
    - per_device_train_batch_size: 1 (or 2, adjusted for Colab memory)
    - gradient_accumulation_steps: 8

    - learning_rate: 2e-4
    - max_seq_length: 512
    - LoRA r (rank): 16
    - LoRA alpha: 32
    - LoRA target_modules: ["q_proj", "v_proj"]
    - Optimizer: paged_adamw_8bit
- **Data Preprocessing:** Instruction-response pairs were formatted into the <s>[INST] Instruction [/INST] Response</s> template required by Mistral Instruct models.
- **Training Duration:** The fine-tuning process took approximately [e.g., 1 hour and 30 minutes] on a Colab T4 GPU for the [Your ~Number] example dataset.
- The resulting LoRA adapter weights were saved and then loaded onto the base model for inference.

5. Challenges & Future Work

- **Challenges:**
    - Curating a large, high-quality fine-tuning dataset was time-intensive.
    - Managing dependencies and occasional resource limitations in the Colab environment required careful setup.
    - Ensuring natural-sounding TTS for all possible LLM outputs requires ongoing pre-processing refinement.
- **Future Work:**
    - Expand the fine-tuning dataset further for even greater nuance and coverage of sleep topics.
    - Implement a more sophisticated dialogue state management system for more complex long-term memory.
    - Experiment with different base LLMs or TTS engines.
    - Conduct more rigorous quantitative and qualitative evaluation of the agent's performance.
    - Develop a more polished standalone application beyond the Gradio interface.

6. Conclusion

This project successfully demonstrates the development of a voice-to-voice sleep coaching agent with specialized capabilities achieved through fine-tuning a Mistral-7B model. The agent can understand relevant queries, provide context-aware advice, and operate within safety boundaries, showcasing the potential of adapted LLMs in specialized domains like sleep health.

Note: Output folder from git-hub repository can be checked for comparative analysis