Checkpoint Report: Final Project

Ankit Mukherjee(50611335), Praveen Kumar Gangapuram(50565977)

April 3, 2025

1 Introduction

This project aims to fine-tune **DistilGPT-2**, a lightweight transformer-based model, to generate **personalized diet plans** based on user inputs. The model is trained using a dataset containing structured diet recommendations. The objective is to enhance the model's ability to generate meaningful responses when given health-related queries.

2 Dataset Preparation

- **Source:** The dataset (`data.csv`) consists of dietary recommendations with three key fields:
  - `Instruction`: Task to be performed (e.g., "Recommend a diet plan").
  - `Input`: User details (age, goals, dietary restrictions, health conditions).
  - `Output`: The corresponding recommended diet plan.
- **Processing:**
  - Converted CSV to JSONL format for Hugging Face compatibility.
  - Mapped each row into a structured format:

```
### Instruction: [Instruction]
### Input: [Input]
### Output: [Output]
```

  - Tokenized the text while ensuring proper padding and truncation (max length: 512 tokens).
  - Split dataset into **90% training, 10% validation**.

3 Baseline Model Training Pipeline

- **Model Exploration:**
  - **Initial Attempt with LLaMA:** We experimented with Meta's **LLaMA** (7B variant) to leverage its larger capacity for natural language understanding. However, due to hardware constraints (limited GPU memory and computational resources), fine-tuning LLaMA proved infeasible. The model's size required optimizations (e.g., quantization, gradient checkpointing) beyond our current infrastructure.
  - **Final Choice:** Switched to **DistilGPT-2** for its lightweight architecture and compatibility with CPU-only training, ensuring stable fine-tuning within resource limits.
- **Pre-trained Model:** Used `DistilGPT-2` from Hugging Face.
  - **Pipeline Setup:**
    - Added a `PAD` token to the tokenizer (since GPT-2 lacks one).
    - Used **Causal Language Modeling (CLM)** with:
      - **Batch size:** 2
      - **Epochs:** 3
      - **Learning rate:** $5e^{-5}$
      - **Weight decay:** 0.01
      - **Evaluation strategy:** Per epoch
  - **Model Output Handling:**
    - Implemented `top-p` sampling and `temperature` scaling for better response diversity.
    - Added **repetition penalty** **n-gram restriction** to avoid redundant outputs.

4 Initial Experiments Results

- The model successfully fine-tuned on the dataset and was able to generate relevant responses.
- Example test prompt used:

```
### Instruction: Recommend a diet plan based on below details
### Input: Age: 25, Goal: Muscle loss, Restrictions: No dairy, Diseases:
### Output:{
  "daily_targets": {
```

```
  "calories": 1678,
  "protein": 127,
  "carbs": 124,
  "fats": 70,
  "fiber": 29,
  "water": "Drink at least 8 glasses of water throughout the day."
},
"meal_plan": [
  {
    "day": 1,
    "meals": [
      {
        "meal_type": "Breakfast",
        "foods": [
          {
            "name": "Scrambled Eggs",
            "portion": "2 large eggs",
            "calories": 180,
            "protein": 14,
            "carbs": 1,
            "fats": 13
          },
          {
            "name": "Whole Wheat Toast",
            "portion": "2 slices",
            "calories": 140,
            "protein": 8,
            "carbs": 24,
            "fats": 2,
            "fiber": 4
          }
        ]
      },
      {
        "meal_type": "Lunch",
        "foods": [
          {
            "name": "Vegetable Stir-fry with Tofu",
            "portion": "1.5 cups",
            "calories": 450,
            "protein": 30,
            "carbs": 40,
            "fats": 20,
            "fiber": 8
          }
        ]
```

```
},
{
  "meal_type": "Dinner",
  "foods": [
    {
      "name": "Oven–Baked Chicken Breast",
      "portion": "4 oz",
      "calories": 200,
      "protein": 40,
      "carbs": 0,
      "fats": 4
    },
    {
      "name": "Broccoli",
      "portion": "1 cup",
      "calories": 55,
      "protein": 4,
      "carbs": 11,
      "fats": 0.6,
      "fiber": 5
    },
    {
      "name": "Brown Rice",
      "portion": "1/2 cup cooked",
      "calories": 110,
      "protein": 2,
      "carbs": 23,
      "fats": 1,
      "fiber": 2
    }
  ]
},
{
  "meal_type": "Snack",
  "foods": [
    {
      "name": "Apple",
      "portion": "1 medium",
      "calories": 95,
      "protein": 0.5,
      "carbs": 25,
      "fats": 0.3,
      "fiber": 4
    },
    {
      "name": "Almonds",
```

```
                "portion": "1/4 cup",
                "calories": 200,
                "protein": 7,
                "carbs": 6,
                "fats": 17,
                "fiber": 4
              }
            ]
          },
          {
            "meal_type": "Second Snack",
            "foods": [
              {
                "name": "Greek Yogurt (Plain, Non-fat)",
                "portion": "1 cup",
                "calories": 100,
                "protein": 18,
                "carbs": 6,
                "fats": 0
              }
            ]
          }
        ]
      }
    ]
}
```


- **Initial Observations:**
  - The model effectively generates structured diet plans.
  - Some responses require refinement to improve accuracy and coherence.

5 Project Management Details

- **Tool Used:** GitHub Projects.
- **Screenshots:**

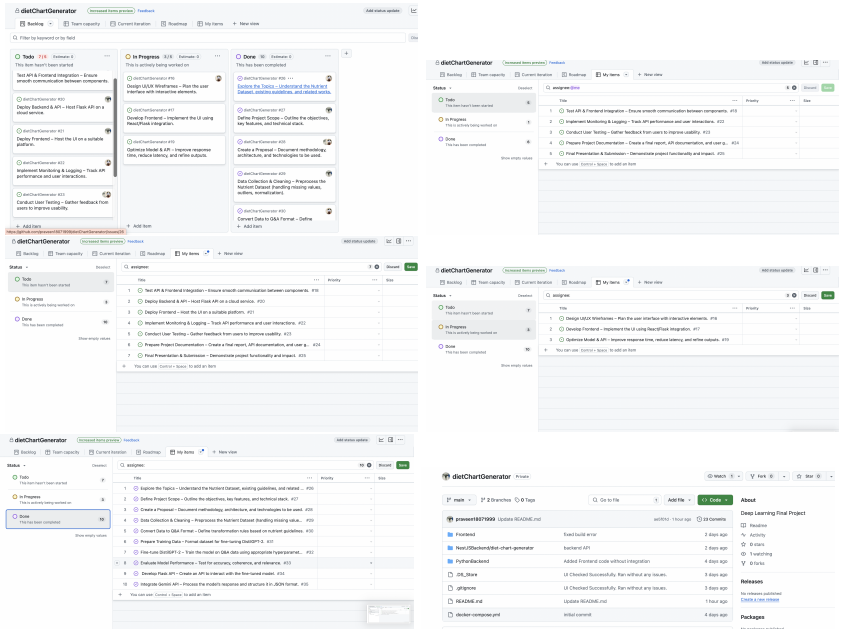

Figure 1: Screenshots of board activities

– **Project Link:** https://github.com/users/praveen18071999/projects/1/views/1

6 Next Steps

1. Develop a **frontend interface** to allow users to input their details and receive diet recommendations in real time.
2. Perform further **hyperparameter tuning** to optimize model performance, focusing on learning rate adjustments, batch size modifications, and training duration.
3. Improve response coherence and diversity by refining **sampling techniques** (e.g., temperature scaling, top-k/top-p filtering).
4. Conduct additional **evaluation** using **perplexity, BLEU score, and human validation** to ensure quality output.
5. Gather **user feedback** on generated recommendations to identify inconsistencies and refine model responses.
6. Implement **logging and monitoring** to track model performance across different inputs.

References

[1] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. *Transformers: State-of-the-art Natural Language Processing*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45. 2020.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108, 2019.

[3] Hugging Face. *Transformers Documentation*. Available at: `https://huggingface.co/docs/transformers`

[4] A. Kumar and R. Patel. *DietGPT: Personalized Diet Recommendations using GPT-2 Fine-tuning*. Journal of AI in Healthcare, vol. 5, no. 2, pp. 34–42, 2023.

[5] Miguel Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2nd Edition, 2018.

[6] Google. *Gemini API Documentation*. Available at: `https://ai.google.dev`

[7] Holtzman, Ari, et al. *The Curious Case of Neural Text Degeneration*. arXiv preprint arXiv:1904.09751, 2019.