Model DDLR: The LLM Fine-Tuning Platform

Introduction

In an era where the customization of AI models has become paramount to address specific user needs and niches, our LLM Fine-Tuning Platform, Model DDLR, emerges as a pivotal solution. This platform is engineered to fine-tune Large Language Models (LLMs) such as GPT models for specialized applications. Whether it's for legal advisories, creative writing, technical documentation, or environmental scientists, our platform equips users with the tools to tailor AI outputs to their precise requirements.

About The Team

We, the development team of Model DDLR, are composed of four dedicated college students from the College of Charleston, embarking on a journey to create a full-stack application. Our project, developed in collaboration with Querri, a local AI-based natural language data platform, aimed to fill a gap in the market by creating a quality dataset editor for large language model training datasets, tailored for ease of use by beginners.

Our team's unique composition brought together diverse expertise and perspectives: Derek Bolch and Rylee Simons, both studying Computer Information Systems, Lindy Mayes, majoring in Computing In The Arts, and Darla Valderrama, a Data Science student. This well-balanced team dynamic enabled us to tackle various aspects of the project with depth and creativity.

Adopting the Agile/Scrum methodology, we enhanced our efficiency and adaptability throughout the development process. Derek took the lead on backend development and integrating the OpenAI API, ensuring our application's robust functionality. Darla focused on database storage solutions and additional backend development tasks, laying the groundwork for our application's data management capabilities. Rylee and Lindy concentrated on the user interface, with Rylee bridging the gap between UI and backend, and Lindy refining the front-end design to ensure an engaging and user-friendly experience.

Through collaborative effort and a shared vision, we succeeded in developing Model DDLR, a testament to our team's commitment to innovation and the advancement of accessible technology.

Core Features

- Custom Model Training: Users can input a project title and a detailed description to

kickstart the creation of a tailored dataset.

- Interactive Dataset Editor: A robust web interface allows users to review, edit, and augment the automatically generated training samples.
- Allows users to upload their own dataset: Users can upload .CSV datasets that will automatically be formatted into our dataset editor for easy adjustments.
- Model Training and Deployment: With just a few clicks, users can initiate the fine-tuning process and deploy their custom models.
- Project Management Dashboard: A centralized dashboard to manage multiple fine-tuning projects, track their progress, and access trained models.
- Security and Privacy: Rigorous data handling and privacy measures to ensure that your training data and models remain confidential.

Getting Started

Prerequisites:

The following must be installed & configured in order to run properly

- Node.js and npm (Node Package Manager)
- MongoDB for database services
- An OpenAI API key

Installation:

1. Clone the repository to your local machine and install the dependencies:

   "git clone https://github.com/ryleesimons/my-app"
   "npm install"

   These two commands will clone the github repository, installing all the necessary files required to run the app

2. Once the repository is installed, we must run a series of commands to install all the necessary dependencies for the app to function properly.

   In your terminal within VS Code or your IDE of choice, run the following commands:

   "npm install openai axios cors dotenv react-router-dom express mongoose bcrypt jsonwebtoken validator multer nodemon --save-dev"

“npm install @mui/material @emotion/react @emotion/styled”

“npm install react-icons --save”

“npm install @mui/icons-material”

Running the Application:

1. To start the front-end server in development mode:
   In your terminal within VS Code or your IDE of choice, run the following command:

   “npm run dev”

   This will automatically navigate to http://localhost:3000 to access the platform.

2. To start the back-end server in development mode:
   In your terminal within VS Code or your IDE of choice, run the following commands:

   “cd backend”
   “npm run dev”

   These two commands will navigate to the backend directory, then initiate the server

Usage Guide:

1. Create a New Project: From the dashboard, initiate a new fine-tuning project by providing a unique project title and a comprehensive description. Alternatively, a user can upload their own dataset into the project to be used in the dataset editor

2. Generate and Edit Training Data: Based on a brief description of what the user wants their fine-tuned LLM to do, Model DDLR uses OpenAI GPT integration to generate samples as a base for your training dataset. The dataset editor allows for meticulous refinement of each sample to the user's preferences.

3. Train Your Model: With your dataset prepared, launch the fine-tuning process. Our

platform handles the rest, leveraging OpenAI's APIs to train your custom model.

4. Deployment and Integration: Once trained, your model is ready for deployment. Integrate it into your application to start benefiting from your tailored AI.

Architecture Overview

- Backend: Developed with Express.js, interfacing with the OpenAI API for all operations related to dataset generation and model training.

- Frontend: A React application providing a seamless user experience for managing fine-tuning projects.

- Database: Utilizes MongoDB to store user projects, training datasets, and model configurations securely.
- Languages: Primarily written in JavaScript, with front-end using HTML/CSS for formatting and style

Acknowledgments

- OpenAI, for providing the GPT models and the API that powers our fine-tuning capabilities.
- Our dedicated team of developers and contributors who have made this project possible.

Contact Us

For any questions or further assistance, please contact us at:

- bolchdp@g.cofc.edu
- valderramadi@g.cofc.edu
- mayesl@g.cofc.edu
- simonsrq@g.cofc.edu

Special Thanks

A special thank you to our sponsors for this project, **Dave Ingram and Tom Schipper of Querri**, for their endless support and advice in this development journey.

We thank you!