**Tweet Emotion Detection using PEFT and Transformer Models**

**Overview**

This project extends tweet sentiment classification using **Parameter-Efficient Fine-Tuning (PEFT)** and **pre-trained Transformer models**. The work is divided into three parts:

* **Part 1**: Data preprocessing and tokenization.
* **Part 2**: Model fine-tuning using PEFT techniques.
* **Part 3**: Evaluation, logging, and inference.

**What is PEFT (Parameter-Efficient Fine-Tuning)?**

PEFT allows fine-tuning large-scale models efficiently by modifying only **small, trainable layers** instead of updating the entire model. This is useful for **memory-constrained environments** and speeds up training while preserving model accuracy.

**Why Use PEFT?**

* **Reduces memory usage**: Instead of updating all parameters, only small layers are trained.
* **Works with Large Models**: Models like **E5-Mistral, Gemma, and LLaMA** can be fine-tuned efficiently.
* **Supports Quantization**: Works well with **BitsAndBytes** for low-precision training.

**Models Used**

This project fine-tunes three **decoder-only** transformer models using PEFT:

* **E5-Mistral-7B-Instruct** (intfloat/e5-mistral-7b-instruct)
  + Based on **Mistral 7B**, optimized for **embedding retrieval** and **instruction-following tasks**.
  + Performs well on **emotion detection** by capturing nuanced text relationships.
* **Google Gemma-2-2B** (google/gemma-2-2b)
  + A lightweight, instruction-tuned model optimized for **efficient fine-tuning**.
  + Works well in **low-resource settings** and is highly optimized for **PEFT techniques**.
* **Meta LLaMA-3.2-1B** (meta-llama/Llama-3.2-1B)
  + A smaller version of LLaMA 3, designed for **text generation and classification**.
  + Uses **causal attention**, making it useful for contextual text understanding.

**Model Comparison**

To evaluate the effectiveness of different fine-tuned transformer models, we compared their performance on tweet sentiment classification:

* **E5-Mistral-7B-Instruct**
  + Best at **sadness (0.56), optimism (0.49), and anger detection**.
  + Lower performance in **trust classification (0.02)**.
* **Google Gemma-2-2B**
  + Strong in **disgust detection (0.46)** and **optimism (0.47)**.
  + Slightly weaker in **surprise prediction (0.03)**.
* **Meta LLaMA-3.2-1B**
  + Best at **love (0.21) and trust detection (0.20)**.
  + Overall weaker compared to E5-Mistral and Gemma in classification accuracy.

**Conclusion**

After evaluating the models on tweet sentiment classification, **E5-Mistral-7B-Instruct emerged as the best-performing model**. It achieved superior accuracy in capturing emotions like sadness, optimism, and anger, making it the most effective model for this task. **Google Gemma-2-2B** performed well in disgust and optimism detection, while **Meta LLaMA-3.2-1B** was slightly behind in overall classification accuracy.

The use of **PEFT** allowed efficient fine-tuning of large-scale models while keeping memory usage low, demonstrating that **decoder-only transformers** like E5, Gemma, and LLaMA can be effectively adapted for tweet emotion detection. This project extends tweet sentiment classification using **Parameter-Efficient Fine-Tuning (PEFT)** and **pre-trained Transformer models**. The work is divided into three parts:

* **Part 1**: Data preprocessing and tokenization.
* **Part 2**: Model fine-tuning using PEFT techniques.
* **Part 3**: Evaluation, logging, and inference.

**Features**

* **Multi-Label Classification**: A tweet can express multiple emotions simultaneously.
* **Parameter-Efficient Fine-Tuning (PEFT)**: Optimizes fine-tuning without updating all model parameters.
* **Pre-trained Transformers**: Fine-tuned models for NLP sentiment tasks.
* **Tokenization**: Uses Hugging Face tokenizers for preprocessing.
* **Training & Evaluation**: Utilizes the Hugging Face Trainer API and logs results using **Weights & Biases (WandB)**.

**Installation**

Ensure all dependencies are installed:

pip install torch transformers evaluate wandb datasets accelerate peft bitsandbytes

**Dataset**

This project uses a tweet-based sentiment analysis dataset, where each tweet is labeled with multiple emotions:

* **Columns:** Tweet, anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust
* **Type:** Multi-label classification
* **Purpose:** To detect emotions in tweets efficiently.

**Model Training**

Each notebook fine-tunes a model with **PEFT** for better efficiency:

* **Part 1**: Prepares and tokenizes the dataset.
* **Part 2**: Fine-tunes the selected transformer model with PEFT.
* **Part 3**: Evaluates model performance and logs results.

Run training with:

python train\_model.py --use\_peft True

**Model Comparison**

To evaluate the effectiveness of different fine-tuned transformer models, we compared their performance on tweet sentiment classification:

* **E5-Mistral-7B-Instruct**: Best in **sadness (0.56), optimism (0.49), and anger detection**. Lower performance in **trust classification (0.02)**.
* **Google Google Gemma-2-2B-2-2B**: Strong in **disgust detection (0.46)** and **optimism (0.47)**. Slightly weaker in **surprise prediction (0.03)**.
* **Meta Meta LLaMA-3.2-1B-3.2-1B**: Best at **love (0.21) and trust detection (0.20)**, but overall weaker than e5 and Google Gemma-2-2B.

🏆 **E5-Mistral-7B-Instruct emerged as the best-performing model**, with **Google Gemma-2-2B as a strong alternative** for specific emotion detection.

**Results**

Evaluation is performed using accuracy, F1-score, and other relevant metrics. Results are logged using **Weights & Biases (WandB)** for tracking experiments.

**Contributing**

Feel free to fork this repository and improve the fine-tuning or experiment with different PEFT settings.

**License**

This project is open-source and available under the MIT License.