Fine-Tuning Tech Support Chatbot

Overview

This project fine-tunes a pre-trained causal language model for a tech support chatbot using Hugging Face's transformers library and PyTorch. The model is trained with gradient accumulation and mixed precision settings to optimize memory efficiency and performance.

Team

We are a team of two, and each person contributes equally to the development and improvement of the chatbot.

Model Architecture

- **Base Model**: Uses AutoModelForCausalLM from Hugging Face
- **Training Settings**:
    - Batch size: per_device_train_batch_size=1
    - Epochs: num_train_epochs=4
    - Gradient accumulation: gradient_accumulation_steps=8
    - Mixed precision: fp16=False
    - Gradient checkpointing enabled for memory efficiency

Training Process

1. Load a pre-trained causal language model.
2. Process dataset into the required format.
3. Fine-tune using the `Trainer` API.
4. Save and evaluate the fine-tuned model.

Running the Training Script

Execute the training script:

```
python train.py
```

Ensure you have the dataset and model properly set before running the script.

Evaluation

The model's performance is evaluated using standard NLP metrics such as:

- Accuracy
- Precision
- Recall
- F1 Score

These metrics help determine how well the chatbot responds to tech support queries.

Deployment

Once trained, the model can be deployed as an API using frameworks like FastAPI or Flask, or integrated into a chatbot interface.

Future Improvements

- Experimenting with different learning rates and optimizers.
- Adding more domain-specific data for better generalization.
- Implementing response ranking to improve answer relevance.

Credits

This project utilizes open-source models and datasets from Hugging Face. Special thanks to the NLP community for making fine-tuning efficient and accessible!