ASEAN LLM League for Impact 2025 - Fine-tuning AI Journey

Process Overview

Data Preparation:

- Use PartyRock to compile a dataset in quiz-style prompts, ensure diversity and relevance to Generative AI, Agent AI, Prompt Engineering, Foundation Model
- Remove duplicate questions and check the relevance, accurracy of responses

Model Fine-tuning:
Applied techniques:

- Key steps included:
  - Data preprocessing (used PartyRock to generate dataset)
  - Fine-tuned, trained model
  - Evaluated and compared results

Performance Evaluation:
Measure accuracy, F1 score and response time, compared to the original and reference models to determine improvement

Key Challenge Faced:

- Insufficent labeled data for fine-tuning
- Do not have so much knowledge about AI, LLM or relevances
- Overfitting during initial training stages

Solutions

- Used **PartyRock** to enrich data
- Leverage **SageMaker** for efficient computation
- Fine-tuning serveral hyperparameters to mitigate overfitting

Achievements

Improved Model Performance:

- Increased accuracy by **59%** compared to the standard model of the organizing committee

Technical Contributions:

- Successfully apply LoRA technique and optimize dataset, improve model inference ability

Competition Ranking:

- Achieved **Top 30** of ASEAN and **Top 3** of Vietnam in competition

Key Learnings

Optimization of Training Pipeline:

- Fine-tuning hyperparameters and optimizing the training pipeline greatly enhanced model efficiency
- Testing multiples strategies to find the best solution.

Practical Experience:

- Participating in the competition provided valuable insights into the challenges and solutions when working with large-scale AI models
- Learn to build future-ready Generative AI applications

Detail About Best Model:

Name Of Best Model: Conqueror-Max2

- Detail about best model: [Best Model Report]
- Several hyperparameters are adjusted:
  - Epoch: 75 - 100 (Make model learn more and deeper)
  - Learning rate: 0.00025 - 0.0003 (Smooth learning process and increase stability)
  - Lora R: 64 - 128 (Fit to size of dataset for learning more with less RAM/GPU)
  - Lora Alpha: 128 - 256 (Often twice as much as Lora)
  - Lora Dropout: 0.2 - 0.3 (Avoid learning by rote, avoid overfitting and generalize)
  - Instruction - Train the model: True (required)
  - Chat Dataset Format: False (required)
  - Per Device Train Batch Size: 4 (Suitable for dataset and other hyperparameters)
  - Per Device Evaluation Batch Size: 4 (Suitable for dataset and other hyperparameters)
- Purpose of each hyperparameters:
  - Epoch: Make model learn more and deeper
  - Learning rate: Smooth learning process and increase stability
  - Lora R: Fit to size of dataset for learning more with less RAM/GPU
  - Lora Alpha: Often twice as much as Lora
  - Lora Dropout: Avoid learning by rote, avoid overfitting and generalize
  - Per Device Train Batch Size: Suitable for dataset and other hyperparameters

- Per Device Evaluation Batch Size: Suitable for dataset and other hyperparameters