Selected Topics in AI Project

Project's Goal: Fine-Tuning Transformer Models for NLP Tasks in Real-World or Simulated Scenarios

Students will design and implement a solution using a Transformer-based model (e.g., BERT, RoBERTa, T5, GPT) to solve a real-world NLP task. The project will include selecting a dataset, fine-tuning a pre-trained model, evaluating performance, and deploying or demonstrating the solution. Teams will also present their work with a live demo and submit comprehensive documentation.

- For the project members teams will be the same as the first discussion.

Phase 1: Problem Definition and Dataset Selection

- Choose a task such as:
  - Text Classification (e.g., sentiment analysis, intent detection)
  - Named Entity Recognition (NER)
  - Question Answering
  - Text Summarization
  - Machine Translation
- Select a publicly available dataset or a custom dataset.
- Preprocess the dataset (tokenization, splitting, formatting for model input).
- Define evaluation metrics (e.g., accuracy, F1-score, BLEU, ROUGE).

Phase 2: Model Selection and Implementation

- Select a Transformer model (e.g., BERT, GPT-2, T5, RoBERTa).
- Justify the choice based on task requirements (e.g., BERT for classification, T5 for summarization).
- Implement the fine-tuning pipeline using Hugging Face Transformers or another framework (e.g., TensorFlow, PyTorch).

- Begin training the model and collect initial results.
- Visualize training progress (e.g., loss curves, validation metrics).

Phase 3: Optimization and Evaluation

- Tune hyperparameters (e.g., learning rate, batch size, max sequence length).
- Apply techniques such as:
  - Data augmentation
  - Transfer learning
  - Early stopping
  - Mixed-precision training
- Evaluate performance using appropriate metrics:
  - Accuracy, Precision/Recall/F1
  - BLEU/ROUGE (for generative tasks)
  - Inference time and memory usage (for optimization)

Phase 4: Reporting and Presentation

- Prepare a final report detailing the Problem statement, Dataset and preprocessing, Model selection rationale, Implementation details, Results and analysis
- Show a demo of the trained model and visual results (graphs, sample outputs).
- Submit all code, data, and documentation

Evaluation Criteria

- **Technical Implementation (40%)**
- **Performance Metrics (20%)**
- **Teamwork and Collaboration (20%)**
- **Documentation (20%)**

Delay penalty(20%): note that any failure to meet any deadlines can result in a penalty, any excesses must be reported beforehand to be accepted, otherwise none will be accepted.

GitHub Repository Requirements

- Use GitHub actively (at least 2–3 commits per member).
- Ensure individual contributions are clearly reflected.
- Avoid plagiarism: Zero marks for forked/copied code without meaningful additions.

Unique Project Approval

- Project ideas must be **unique** and approved by the instructor in Phase 1.
- Duplicate projects will be rejected unless a clear distinction is shown (e.g., using different models, languages, or datasets).