Mini Project 1: Language Models and Agents

Deadline: September 15, 2025

1 Goal

The aim of this mini-project is to build a small autoregressive language model (LM) for multiple languages from scratch. This is an individual project where you will handle the entire workflow:

- Data collection and creation.
- Pretraining the model.
- Fine-tuning for specific tasks.

The provided tasks will focus on simple reasoning abilities such as spelling correction, word formation, and basic logic-based text generation.
You can find the **language id, model id and fine-tuning task** assigned to you here and here.

2 Project Phases

2.1 Phase 1: Data Collection and Preprocessing

- Collect corpus for three languages:
  - English – 50% of total tokens.
  - Mother tongue – 30–40% of total tokens.
  - Indian language – 10–20% of total tokens.
- Minimum total tokens required: 3 billion.
- Apply preprocessing techniques such as cleaning, deduplication, sentence segmentation, and train/validation/test split.
- Submit scripts and token statistics.

2.2 Phase 2: Tokenizer Training

- Train a tokenizer (BPE, SentencePiece, or similar).
- Ensure that the vocabulary supports all three languages adequately.
- Submit tokenizer files and evaluation of token coverage.

2.3 Mid Evaluation

Phases 1 and 2 deliverables along with a report containing work done so far, timeline, and steps for future phases will be used for evaluation and feedback.

2.4 Phase 3: Model Pretraining

- Use the given model type with a parameter size between **100M–150M**.
- Train an autoregressive LM on the processed corpus using any language/framework such as PyTorch or TensorFlow.
- You may use free-tier compute platforms such as Kaggle, Google Colab, etc.
- Implement checkpointing in batches to manage long training times and resource limits.
- Submit pretrained model checkpoints and training logs.

2.5 Phase 4: Finetuning on Reasoning Tasks

- Finetune the pretrained model for two reasoning tasks (provided in the sheet earlier).
- Use chat-style templates for data preparation.
- **Do not make the model memorize the reasoning tasks**—this will be checked during evaluation, and penalties will apply if found.
- Submit fine-tuned models and scripts.

2.6 Phase 5: Evaluation and Analysis

- Evaluate pretrained and fine-tuned models using appropriate metrics (e.g., perplexity, accuracy).
- Perform detailed error analysis and discuss your observations.

3 Documentation & Deliverables

1. Language corpus and preprocessing scripts.
2. Tokenizer files.
3. Mid-evaluation report with:
   - Work done so far.
   - Timeline and steps for Phases 3–5.
4. Training and evaluation scripts.
5. Pretrained model checkpoints.

6. Fine-tuned model checkpoints.

7. Final report with approach, methodology, experiments conducted, and result analysis.

4 Constraints & Notes

- Training must be completed in batches with checkpoints.
- No extension requests will be entertained.
- Start early and verify your data and approach before training due to time and resource constraints.
- You may use available GPU/TPU resources but ensure reproducibility of results.
- All the relevant data sources must be cited.

5 Submission Format

- Code and scripts used for preprocessing, training, and evaluation.
- Models and checkpoints uploaded to GitHub or Hugging Face Hub.
- **Mid-evaluation**: PDF report with work done so far, timeline, and steps for Phases 3–5.
- **Final-evaluation**: PDF report with approach, methodology, experiment details, and results.
- Submissions will be taken through Github Classroom here.