Conversational Response Prediction System

This report documents the process of developing an offline Conversational Response Prediction System capable of generating context-aware replies for two-person chat data. The system leverages GPT-2, a Transformer-based language model, trained and evaluated using a structured NLP pipeline.

1. Data Preprocessing and Tokenization

The dataset was grouped by conversation IDs and sorted chronologically. Text messages were cleaned, normalized, and separated by users (User A and User B). Special tokens [SEP] and [RESPONSE] were added to improve context segmentation. A maximum of 5 prior messages formed the context window for predicting the next response.

2. Model Fine-Tuning

The GPT-2 model was fine-tuned offline using the Hugging Face Transformers library. Key hyperparameters included: Epochs: 3 Batch size: 2 Learning rate: 5e-5 Optimizer: AdamW Training was executed on a CUDA-enabled GPU. Validation loss decreased steadily across epochs, indicating effective learning.

3. Response Generation

The trained model generated responses based on unseen test contexts. While limited data caused occasional incoherence or repetition, the model captured basic conversational structure and user response patterns.

4. Evaluation Metrics

Model performance was measured using BLEU, ROUGE, and Perplexity metrics: BLEU Score: 0.0102 ROUGE-1: 0.1500 ROUGE-2: 0.0000 ROUGE-L: 0.1000 Perplexity: 53.9670 Although results were modest due to small data size, the model demonstrates the feasibility of response prediction using Transformers.

5. Model Choice, Optimization, and Deployment

Model Choice: GPT-2 was selected for its autoregressive text generation capabilities, strong context understanding, and moderate hardware requirements.
Optimization: Mixed precision, learning rate scheduling, and gradient clipping were used for efficient training.
Deployment: The final model was saved in both safetensors and joblib formats for flexible reuse. Joblib enables quick loading for deployment via APIs (Flask/FastAPI) or integration into chatbots.

Conclusion

This project presents an end-to-end conversational AI pipeline — from preprocessing and model training to evaluation and deployment. Future improvements include training on larger corpora, integrating attention visualization, and applying reinforcement learning for human-like dialogue adaptation.