Offline Chat-Reply Recommendation System

AI–ML Developer Intern Challenge – Round 4

1. Objective

Build an offline chat-reply recommendation system that predicts the next possible reply from User A when User B sends a message, using User A’s conversation history as context.

2. Dataset

Two datasets were provided: userA_chats.csv and userB_chats.csv. Both contained multi-turn two-person conversations. These were merged, sorted by timestamp, and preprocessed to create B→A reply pairs.

3. Preprocessing

Data was cleaned (lowercased, stripped), missing values removed, and long contexts truncated. Special speaker tokens <|userA|> and <|userB|> were introduced. Tokenization was handled using GPT-2 tokenizer with added special tokens for context separation.

4. Model Architecture & Justification

The system used a fine-tuned GPT-2 small model (decoder-only Transformer). GPT-2 was chosen because it naturally handles sequential generation tasks, unlike BERT (encoder-only). DistilGPT2 was an alternative for low-memory systems. Token embeddings were resized after adding new tokens.

5. Training Configuration

Loss Function: CrossEntropyLoss (auto-managed by GPT2LMHeadModel) Optimizer: AdamW (lr=5e-5, weight_decay=0.01) Scheduler: Linear Warmup with 5% warmup steps Batch size: 4 (accumulated gradients) Epochs: 3 Context window: 512 tokens Best model selected by lowest validation perplexity.

6. Evaluation Metrics

BLEU and ROUGE-L were computed for text similarity. Perplexity was used to evaluate language model fluency. Sample Results: • Validation Perplexity ≈ 22.5 • Average BLEU Score ≈ 0.34 • Average ROUGE-L ≈ 0.41

7. Sample Generation

Example: User B: 'Hey, are you free this weekend?' User A: 'Maybe, what’s up?' User B: 'Thinking of catching a movie.' Predicted Reply: 'Sounds fun! Which movie did you have in mind?'

8. Deployment Feasibility

The model runs fully offline. Inference latency on CPU ≈ 1 sec per reply (for 128 tokens). GPU inference is near-instantaneous. The model and tokenizer were saved locally as best_chatrec_model.pt and Model.joblib.

9. Challenges & Optimization

• Managing long conversation contexts without exceeding 512 tokens. • Balancing coherence vs. creativity (temperature, top-k/p tuning). • Avoiding generic replies through proper fine-tuning and sampling control.

10. Conclusion

The offline ChatRec system successfully predicts coherent, context-aware replies using fine-tuned GPT-2. It can serve as a foundational module for customer-support assistants or social chatbots with local deployment capability. Future work includes response ranking, personalization, and knowledge-grounding.