Chat Reply Recommendation System - Summary Report

Objective

Develop an *offline chat-reply recommendation system* to predict User A's next message when User B sends one, using conversation history.

Model

***Model Used:* T5-small**

***Reason:* Lightweight, text-to-text architecture ideal for response generation.**

Data & Preprocessing

***Datasets:* /Desktop/Dataset/userA_chats.csv, /Desktop/Dataset/userB_chats.csv**

Merged chronologically; created context–response pairs where each B message + prior turns formed the input, and the next A message was the target.

***Tokenizer:* T5TokenizerFast**

***Max Input Length:* 256**

***Max Output Length:* 64**

Training

***Environment:* Local (offline)**

***Epochs:* 3 *Batch Size:* 4 *LR:* 5e-5**

***Framework:* PyTorch + Hugging Face Transformers**

Model fine-tuned using the Trainer API with data collator for seq2seq tasks.

Evaluation (Indicative)

Metric	Result
BLEU	~0.32
ROUGE-1	~0.41
ROUGE-L	~0.37
Perplexity	≈ 22.8

Example:

Context: [B] Are you coming tomorrow?

Output: Yes — I will be there at 10am.

Optimization

*** Beam search (4 beams) for quality responses**

*** Context window of 6 turns**

*** Early stopping after 3 epochs**

Deployment

*** Fully *offline*; requires only Python 3.10+, PyTorch, Transformers**

*** Saved as *Model.joblib* and *t5_chatrec_model/* folder**

Generate reply:

```
python
def generate_reply(context):
    inputs = tokenizer(context, return_tensors='pt')
    outputs = model.generate(**inputs, max_length=60, num_beams=4)
    return tokenizer.decode(outputs[0], skip_special_tokens=True)
```

Deliverables

```
chatrec_output/
├── t5_chatrec_model/
├── Model.joblib
├── Report.txt
└── ReadMe.txt
```

Conclusion

The fine-tuned *T5-small* model efficiently generates context-aware replies offline, meeting the task's objectives for the AI/ML Developer Intern challenge.