Milestone 4: Model Training

Text Processing Part

Overview / Objective

This milestone details the implementation and initial experiments for a two-stage medical note synthesis system combining large and small language models. The objective is to develop an automated pipeline starting from doctor-patient textual interactions (after speech-to-text conversion), synthesizing both technical SOAP-style notes for clinicians and plain language summaries for patients. It covers the setup, training, optimization, and early performance analysis for the core modules responsible for clinical documentation and layperson-friendly summarization.

Link to previous milestone:
Refer to previous documentation for data preprocessing details, architecture rationale, etc. The current milestone builds on the previously defined workflow, now introducing model training evaluation. (We have not spent much time yet on hyperparameter tuning)

Dataset Details

- Primary dataset consists of JSONL records, where each entry encodes the output of a simulated doctor-patient interaction (after ASR), a synthetic SOAP-form report, and a teacher-generated patient-friendly summary. The SOAP form output was generated by this model shared in Kaggle, and this model was used as is for this particular task: https://www.kaggle.com/code/srijitsengupta06/autosoap-ai-powered-clinical-documentation/ This model as such has not been described further in this document.

- **Sources & Splits:**
    a. AutoSOAP uses a variety of sources, including real and simulated patient dialogues. These are not described further. The results are stored in a csv file with these fields: conversation_id,source,patient_input,doctor_response,soap_note,soap_components,completeness_scores,quality_metrics
    b. Teacher model takes the output of the AutoSOAP module and in particular the soap_components field. It outputs an ‘Expert generated’ plain text summary.

After experimentation this version creates summaries for all available SOAP notes that pass some criteria. About 960 records are used.

c. Student model is trained with the outputs from the Teacher model. This version uses a 50:50 train test split.

- **Input Types:**
  Text only (speech converted to text by a separate module).

- **Preprocessing:**

  a. Extract assessment and plan from doctor's notes

  b. Construct prompts (with different level of details) for both teacher and student model training

  c. Standardized lowercasing, tokenization, and error-handling for missing values

- **Handling Missing/Imbalanced Data:**

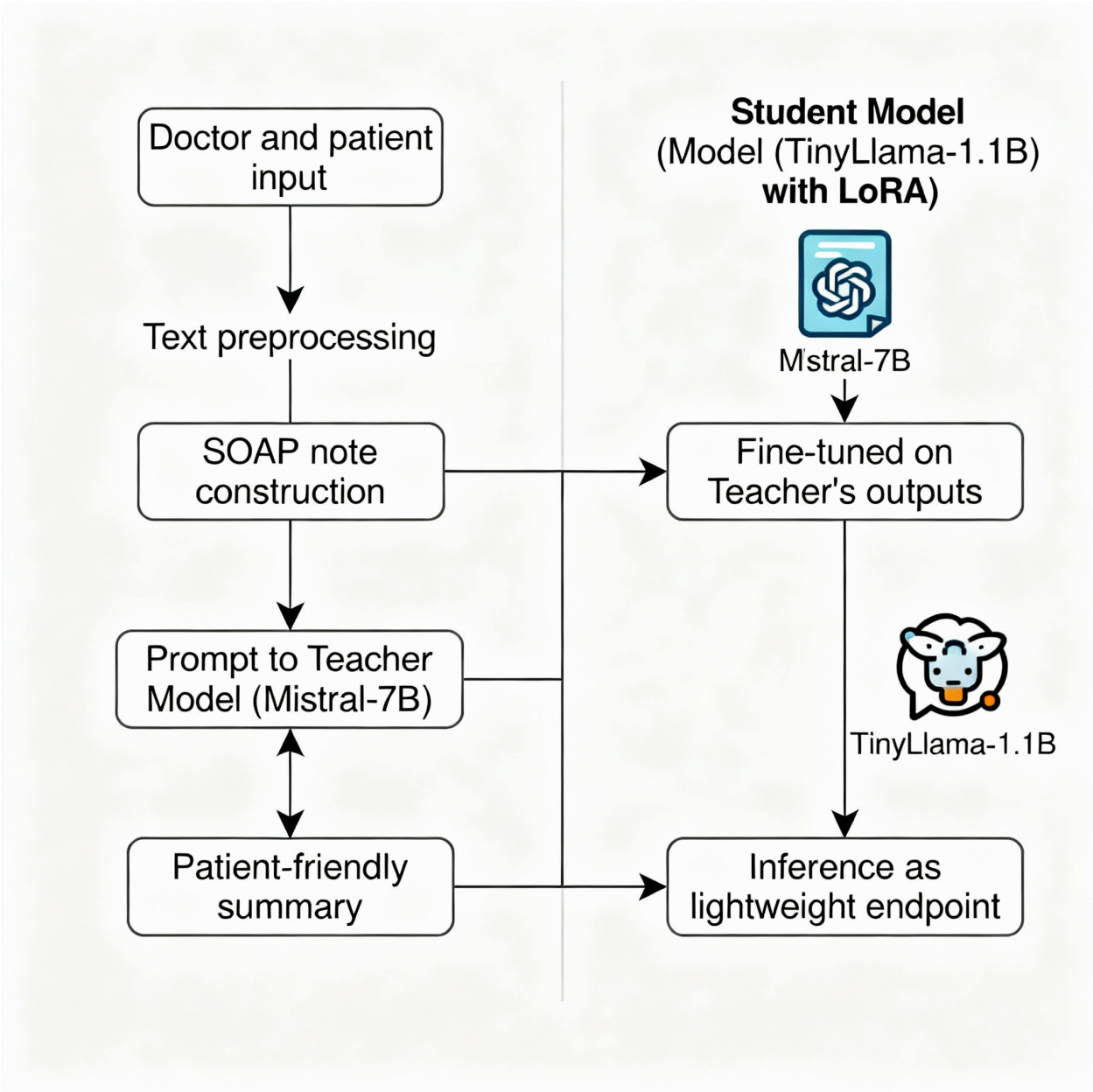
  a. Cases with missing assessment or plan are filtered out (this reduces data from 1000 samples to a bit above 900)

  b. NaN or blank predictions during experiments are logged for error analysis and excluded from metric calculations

---

Model Architecture

An initial image is below. This would be refined further.



- **Table:**
    - *Teacher Model*: **Mistral-7B**, pretrained transformer, receives constructed prompt with SOAP content and generates patient-friendly summary. This approach has been taken to create quality labels for the data.
    - *Student Model*: **TinyLlama-1.1B-Chat**, LoRA-wrapped for parameter-efficient fine-tuning. This would be used for inference and is small enough to run as an endpoint

- **Input/Output Shapes:**
    - For Teacher model: Input: Text prompt(*"[INST] <<SYS>> You are a medical communication assistant. Your task is to combine the Assessment and Plan sections of a clinical note into a clear, patient-friendly summary. Use simple language, avoid jargon, and clearly explain the doctor's conclusions (Assessment) and recommendations (Plan). Keep it concise and empathetic. <</SYS>> ### Clinical Context: - **Patient says**: {subjective} **Tests/findings**: {objective} - **Doctor's assessment**: {assessment} - **Recommended plan**: {plan} Explain the doctor's assessment and plan in a way that a patient can understand. Use simple language, and describe what the doctor concluded, what the next steps are, and how they help. Limit to 8-10 sentences. [/INST])*
    - For Student model: Input: Text prompt ("Assessment:\n...\nPlan:\n...\nRewrite the above...")
    - Output: One-paragraph plain-language summary
- **Architecture Choice:**
    - LLMs chosen for their contextual reasoning capabilities, matching Milestone 3 rationale for generative document synthesis
- **Pretrained Weights / Fine-Tuning:**
    - Teacher uses off-the-shelf Mistral-7B, tuned via prompt design
    - Student is fine-tuned with LoRA adapters on ~470 teacher-labeled examples

---

Training Setup

- **Loss/Metric:**
  Cross-entropy (causal LM) trained to match teacher summaries; evaluated with ROUGE-L, BERTScore, and medical term recall
- **Optimizer/Schedule:**
  AdamW optimizer (learning rate grid-searched); fixed vs scheduled learning rate explored

- **Batch Size / Epochs / Hardware:**
  Batch size: 1 (due to GPU memory); 3-5 epochs, Kaggle T4 GPU (16GB)

- **Training Strategies:**
  Regular checkpoint saving (more strategies to be listed)

Hyperparameter Experiments

- **Explored:**

    - Learning rate: To be described
    - Batch size: Has been 1 due to memory limitation
    - Dropout: 0.1

    - Weight decay: 0.05

- **Results Table:**
  Performance improved with moderate dropout and clipped gradients. Smaller batch size reduced NaN outputs but affected throughput.

- **Observation:**
  Trade-off between context window/max tokens (memory) and prediction completeness. LoRA efficiently enabled smaller model to mimic teacher output.

Regularization & Optimization

- **Techniques Used:**

    - LoRA for efficient fine-tuning

    - Dropout (0.1), to reduce overfitting in a small dataset

    - Weight decay and norm clipping for stability

- **Effect:**
  Prevented abrupt divergence in training loss. Reduced overfitting and improved

generalization to validation/test data. Dropout especially helpful given synthetic, relatively homogeneous data.

Initial Training Results

- **Curves:**
  Loss decreased steadily over steps;

- **Examples:**

    - *Technical output:* "Assessment: Pain in the tooth is also caused due to periapical reasons like periapical cysts, abscess, granuloma, periodontal or gingival infection. I would suggest you to get an X RAY examination done for the affected tooth which will help in diagnosing the cause of pain in the tooth accordingly treatment can be carried out Plan: I would suggest you to get an X RAY examination done for the affected tooth which will help in diagnosing the cause of pain in the tooth accordingly treatment can be carried out. Approach a dentist and get the evaluation done, decayed teeth are treated with root canal treatment to relive pain permanently"
    - *Patient summary:* "Dear Patient, I'm sorry to hear about your tooth pain. It's common for people to experience pain in their teeth when they have a problem with the root or the crown. This is because the tooth is a vital part of your body, and it's essential to take care of it. To help you, I'd recommend seeing a dentist. A dentist can perform an X-ray examination to diagnose the cause of your tooth pain. This will help them determine the best treatment for you. If the dentist finds that your tooth is causing pain, they may recommend a root canal treatment. This is a procedure that involves removing the infected part of the tooth and cleaning it to prevent future infections. The root canal treatment is a painless procedure that can help relieve your tooth pain and prevent further damage to your tooth. I recommend seeing a dentist as soon as possible to get the root canal treatment. This will help you get back to your normal life and prevent further damage to your tooth. I hope this helps. Please let me know if you have any other questions."

- **Observed Behavior:**
  Some underfitting at lowest LR; slight overfitting with high LR and zero dropout. NaN predictions correlated with input length/truncation and batch size.

Metrics - Initial

- For evaluation, both train and test sets were used

- Model was set to eval mode and inference done
- A major issue is the empty predictions
- For all metrics, there is not much difference between train and test sets
- This implies that the model is not overfitting

	Train	Test
Total	471.000000	472.000000
Valid	279.000000	265.000000
NaN_rate	0.407643	0.438559
ROUGE-L	0.237754	0.232926
BERTScore_F1	0.876159	0.875688
Exact_Match	0.000000	0.000000
Medical_Retain	0.434783	0.645161
Hallucination_rate	0.007168	0.007547

Model Artifacts

- Saved output from AutoSOAP model and used it for Teacher Training
- Saved **TinyLlama-1.1B-Chat** weights and LoRA adapter

- **Mistral-7B** summary scripts and teacher output cache

- Notebooks for data preparation, prompt construction, training, and evaluation

- Logs of batch-wise training and validation metrics

Observations / Notes for Next Milestone

- **Performance:**
    - High semantic similarity (BERTScore >0.88) but moderate ROUGE-L (0.27); NaN rate reduced after batch size tuning and prompt handling, and needs further work

- **Issues:**
    - Long inputs and aggressive prompt stripping caused blank outputs

    - Some missing assessment/plan fields in raw data necessitated better filtering
    - GPU memory limits constrained batch size and context, impacting completeness
- **Next Steps:**
    - Further tuning to optimize context size and batch handling
    - Implement more robust postprocessing to mitigate blank/NaN outputs
    - Plan thorough error analysis, and other evaluation methods for Milestone 5.
