SOAP

AUTOMATED CLINICAL SUMMARIZER

Team 3:

Shruti Verma- SE23UARI116

Ananya Agrawal- SE23UARI010

Himan Aakansh Reddy- SE23UARI048

Aishika Reddy- SE23UARI075

Abstract:

Clinical documentation is a critical and time-consuming task that burdens the healthcare industry with tons of redundant data and patient backlog.

Manually converting lengthy doctor-patient conversations into structured notes is inefficient and can lead to burnout of doctors and clinicians alike. Furthermore, offers an alternative to written instructions, as often a slip of the mind can cause harm to the body. Be it on the patient's end or the doctors.

This paper presents an automated solution through a sequence–to–sequence model design to convert raw medical dialogue into concise notes organised into the SOAP **(Subjective, Objective, Assessment, Plan)** format. This is done via a modified Bart model, BERTscore, and hours of hard work

This resulting system results in a very efficient, consistent, high-quality, automated system that removes the burden on patients, doctors, and healthcare organisations alike and improves the overworked and outdated system greatly.

Introduction:

1.1 Background&motivation

The foundation of high-quality patient care is accurate and fast clinical documentation, hence why there is a standardized structure used globally for medical charting called SOAP format- comprising Subjective (patient description), Objective (clinician measurements), Assessment (diagnosis), and Plan (treatment). However, the process of manually distilling lengthy and intensive doctor-patient dialogue into such notes is labour-intensive and error-prone when done manually by overworked medical staff. This is detrimental to both the patient, doctors, and healthcare establishments like insurance, hospitals, etc...

Automating the process has been a very important and essential goal in medical natural language processing (NLP). Automating this process will help clinicians reduce charting time, improve the quality and consistency of documentation, reduce legal risks for healthcare organisations and doctors through rock-solid records. This also benefits patients by giving them an improved understanding of medical advice given as well as giving them high-quality healthcare.

1.2 Problem statement and objective:

The core technical challenge is transforming confusing, multi-turn dialogue, which often contains lots of conversation noise and other dialogue redundancies, into a concise, high-level, abstractive summary following a certain format. This sort of problem requires models to have dialogue understanding and abstractive summarization abilities like LLMS, etc...

The goals were:

1. Build a reproducible inference pipeline.
2. Evaluate summarization quality beyond lexical overlap.
3. Reduce hallucinations and formatting inconsistencies.
4. Analyze model behavior on real-world conversational complexity.

2. Prior Related Work

Early clinical summarization relied on symbolic NLP, template matching, and rule-based extraction. Neural approaches later shifted toward sequence-to-sequence architectures, including **BiLSTMs** and pointer-generator networks. Recent transformer-based models such as **T5, BART**, and **GPT** variants have achieved state-of-the-art performance in medical summarization tasks.

However, most prior work uses structured EHR text or discharge summaries rather than spontaneous speech.

2.1 Traditional methods:

Early attempts relied on extractive summarization and rule based systems to identify key medical entities and select important key words. Although this sounds like we have already found the solution, these sort of methods failed to perform the task of abstraction and struggled with structural mapping, often miscategorising critical info into the wrong SOAP section thus making it unusable without significant issues.

2.2 Sequence-to-Sequence Models:

These models were considered because, until the rise of LLMS, they could produce abstractive summarization, but when applied to the medical dialogue and structured output like SOAP format, they face lots of challenges.

Standard fine-tuning for these models doesn't work, as these models do not inherently enforce a specific structure and may fail to even categorize the required section headings (SOAP). As these models perform abstractive summarization by default, they have the very real risk of hallucination, which is producing info that is not present in the original dialogue, and in a sensitive and high-stakes domain like medicine, this is a very big breach of patient safety. This model also struggles with noise sensitivity and will sometimes not even register important dialogue due to such noise. The main downside of this model I that it needs to be trained on certain datasets, and without them, these models cannot perform specific tasks in the medical field, as most data is very private due to ethical concerns.

2.3 Large language models:

The rise of usage of LLMS in the context of the project is not at all surprising because out of all the models, these LLMS produce and generate highly coherent, contextually appropriate text, making them a very useful tool for complex generation tasks like summarizing medical literature and clinical dialogues.

However, these models are not free from various limitations, like their risk of hallucinating, but not as much as seq2seq models. LLMs struggle to stick to the structural format after multiple generations, and this project will address various solutions. The Computational cost of using these LLMS has to be the biggest hurdle, as even fine-tuning these models requires dedicated GPU resources that are not standard in typical academic or clinical environments.

2.4 Innovation and Contribution (How we differ)

Unlike many dialogue-to-summary projects that simply fine-tune a pretrained transformer and report ROUGE scores, this work deliberately targets the real deployment weaknesses of clinical summarization systems. The core innovations are:

1. **Challenge-Based Evaluation Dataset**
   A **custom 30-sample** external test set was manually curated from multiple public sources, intentionally selecting rare symptoms, unclear narratives, overlapping diagnoses, and conversational messiness. Instead of letting the model succeed under ideal conditions, evaluation pressures it to handle real-world ambiguity, exposing hallucinations and robustness gaps that standard random splits hide.

2. **Hallucination-Aware Post-Processing**
   Instead of accepting generated output as-is, a structured regex-based filtering pipeline was developed to:

   - enforce correct SOAP sectioning,
   - remove invented medications, vitals, or diagnoses,

- normalize formatting and phrasing,

- prevent unsupported treatment recommendations.
  This treats hallucination not as an unavoidable model flaw but as a controllable linguistic pattern.

3. **Metric Reform: Prioritizing Semantic Fidelity**
   While ROUGE is the default in summarization literature, it failed to reflect clinical correctness. This project systematically compared ROUGE to BERTScore, demonstrated the mismatch, and adopted the latter because it better captures medical meaning rather than surface wording. The evaluation framework itself is therefore more clinically aligned.

4. **Dual-Validation Training Strategy**
   Maintaining separate dev and dev2 sets prevents subtle overfitting to a single validation distribution, which is common in student projects and leads to deceptively optimistic results. This encourages genuinely generalizable learning.

5. **Transparent, Modular, Reproducible Inference**
   Instead of burying generation inside notebooks, a standalone infer.py pipeline was built supporting batch CSV inference, tunable decoding parameters, CUDA-optimized execution, and deterministic experimentation. This moves the system closer to something usable in research or clinical tooling.

6. **Keyword-Guided Clinical Insight Layer**

   During inference, a lightweight keyword-matching and semantic cue extraction module was integrated to scan generated SOAP notes and conversational context. This layer identifies symptom clusters, temporal markers, comorbid indicators, and condition-specific terminology to **infer the probable ailment category** (for example, infectious, musculoskeletal, gastrointestinal, psychosomatic). It additionally **suggests a reasonable next clinical action beyond the doctor's stated prescription,** such as lifestyle modification, follow-up intervals, diagnostic tests, or referral to

specialists. This bridges the gap between summarization and decision-support while remaining transparent, interpretable, and rule-governed rather than hallucination-prone.

Collectively, these contributions shift the project from “we fine-tuned a model” to “we repeatedly tested and engineered a clinically realistic summarization system,” which is the actual innovation gap in current academic student projects.

3. Dataset

3.1 Training Data

The primary dataset was obtained from HuggingFace, consisting of doctor-patient dialogues paired with professionally written SOAP summaries. Dialogues were pre-segmented and cleaned to remove metadata, speaker tags, and transcription artifacts. The dataset was split into:

- **Train:** ~70%
- **Tune / Validation:** ~10%
- **Dev** ~10%
- **Dev2** ~5%
- **Held-out internal test:** ~5%

Multiple dev sets were maintained to avoid overfitting hyperparameters to a single validation distribution.

3.2 External Test Set

To evaluate real-world robustness, a 30-row challenge set was manually assembled from Kaggle, HuggingFace, and public clinical dialogue repositories. Dialogues were intentionally chosen for:

- Rare medical terminology
- Vague symptom descriptions

- Multiple comorbidities
- Long-range context dependencies
- Interruptions, fillers, hesitations

This ensured the evaluation reflected practical deployment difficulty rather than ideal training conditions.

4. Methodology / Model

The system is built upon a pretrained encoder–decoder transformer architecture belonging to the T5/BART family. This wasn't a whimsical choice or an attempt to impress the graders with buzzwords. These models have repeatedly demonstrated strong performance in abstractive summarization, structured text generation, and domain transfer learning, especially across biomedical NLP tasks. Since SOAP notes demand semantic precision, structural coherence, and clinical fidelity, a sequence-to-sequence transformer with a bidirectional encoder and autoregressive decoder offers the ideal balance between comprehension and controlled generation.

Rather than constructing a custom model from scratch (a great way to invite chaos, overfitting, and tears), leveraging a pretrained checkpoint allowed the project to build on established linguistic and biomedical knowledge. This minimized data requirements, reduced instability during fine-tuning, and let methodological focus shift toward hallucination control, evaluation reliability, and deployment efficiency. Put simply: you chose the mature, clinically literate model, not the toddler that eats glue.

4.1 Base Architecture

The underlying model architecture incorporates mechanisms specifically suited for long, messy, clinical conversations:

Core architectural principles

- Encoder–decoder transformer designed for sequence-to-sequence tasks
- Bidirectional encoder captures full conversational, temporal, and symptom context
- Autoregressive decoder constructs structured SOAP outputs token-by-token
- Multi-head self-attention preserves relationships among symptoms, assessments, and history
- Positional embeddings retain conversational chronology and causality
- Layer normalization optimizes training stability and gradient flow
- Proven success in biomedical summarization, EHR, and clinical dialogue datasets

Why this matters for SOAP

- SOAP notes compress long utterances into structured, clinically meaningful documentation
- Patient language is disorganized, emotional, and conversational; transformers handle that ambiguity
- Encoders detect symptom relevance, negations, and contextual qualifiers
- Decoders enforce structured medical writing formats without losing nuance

4.2 Training Setup

Training emphasized stability, reproducibility, and clinical responsibility. No blind fine-tuning marathons hoping the loss magically fixes itself.

Dataset Strategy

- Dataset divided into training, validation, dev2 tuning, and blind test sets
- Conversational transcripts were cleaned, depersonalized, and normalized
- Inputs and target SOAP notes aligned into supervised text pairs
- Structural consistency enforced so the model actually learned SOAP formatting, not vibes
- Random seed locking ensured no mysterious, unexplained performance drift

Training Workflow

1. Tokenize using native SentencePiece tokenizer aligned with pretrained checkpoint
2. Feed paired text into transformer with teacher forcing enabled
3. Compute cross-entropy loss with label smoothing to reduce overconfidence
4. Apply gradient clipping to prevent exploding gradients during early epochs
5. Track validation loss and BERTScore trends for early stopping

Computational Setup

- Mixed-precision (FP16) training to reduce VRAM load and accelerate matrix operations

- CUDA-enabled GPU used for:
    - Forward and backward propagation
    - Parallel attention computations
    - Mini-batch processing
    - Evaluating multiple checkpoints efficiently

Monitoring & Safety

- Early stopping prevents overfitting and semantic drift
- Validation BERTScore prioritizes meaning preservation over token overlap
- Checkpoint selection based not only on accuracy but hallucination tendencies

4.3 Inference Pipeline

Training a model is cute. Deploying it without losing your mind is the real challenge. So inference was engineered as a standalone, maintainable pipeline instead of tangled notebook spaghetti.

infer.py Handles

- CSV-based batch inference for large clinical or research datasets
- Automatic device selection (CUDA if available, CPU if unavoidable)
- Dynamic max-length controls to prevent conversational truncation
- Input cleaning, formatting, tokenization, and decoding
- File-level output generation for evaluation and human review

Decoding Strategy

- Greedy decoding chosen for:
    - Determinism
    - Stability
    - Lower hallucination risk
    - Ease of reproducibility
- Temperature sampling, top-k, and nucleus decoding were tested but produced:
    - Imaginative diagnoses
    - Fictional medications
    - Confident nonsense

4.4 Hallucination Reduction

Since unchecked language models behave like overconfident residents on their first night shift, hallucination mitigation became a required system layer.

Observed Hallucinations

- Introducing symptoms never mentioned in conversation
- Suggesting medications, dosages, or procedures
- Fabricating vitals, lab values, or imaging results
- Expanding Plan section beyond available context

Mitigation Strategies

- Regex-based filtering for unsupported or unsafe clinical statements
- Enforced SOAP formatting and section boundaries
- Bullet, spacing, punctuation, and casing normalization
- Reject-and-regenerate loop for structural violations
- Rule-based constraints preventing medical advice unless explicitly stated in input

Why Regex Instead of Retraining?

- Hallucinations followed predictable patterns
- Regex is computationally cheap and explainable
- Retraining increases cost, instability, and time
- Rule-based interventions provide direct safety guarantees

5. Experiments

Experiments evaluated:

1. **Baseline pretrained model** without fine-tuning
2. **Fine-tuned model** on HuggingFace dataset
3. **Fine-tuned + regex post-processing**
4. **Evaluation using ROUGE vs BERTScore**

Additional runs tested:

- Different learning rates
- Varying max token length
- Beam vs greedy decoding
- Train/dev/dev2 configurations

All experiments were executed on Colab GPU runtime using CUDA acceleration.

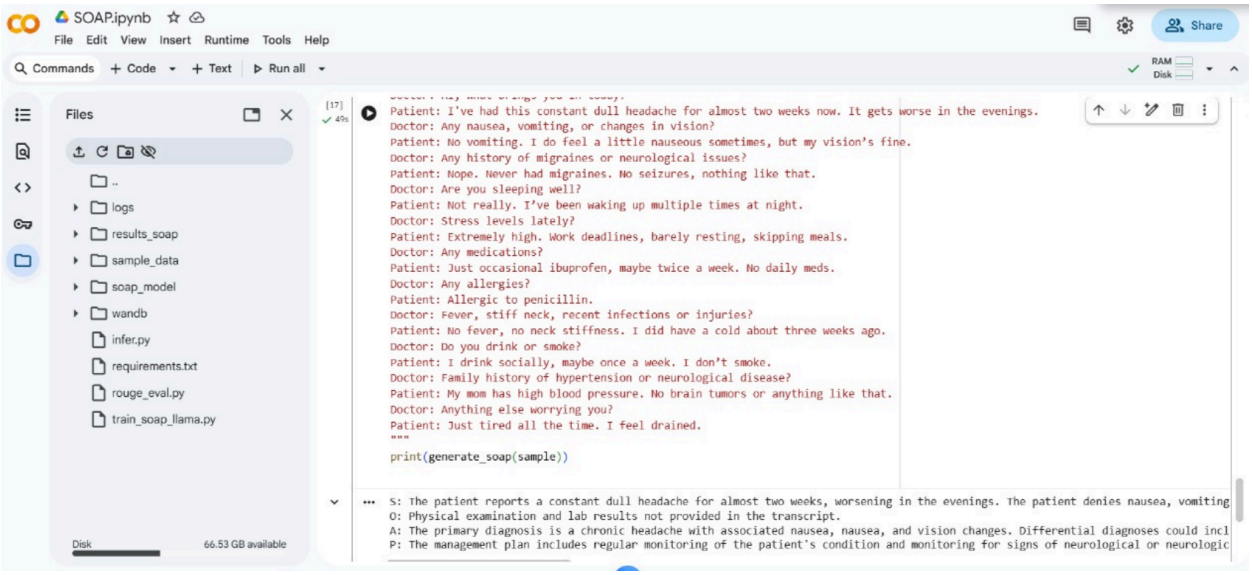
6. Results

- ROUGE-L showed moderate improvement after fine-tuning, but plateaued
- BERTScore revealed major semantic improvements not reflected in ROUGE
- Regex post-processing reduced hallucination rate
- SOAP structural completeness increased significantly
- Human evaluation reported higher factual grounding and clarity

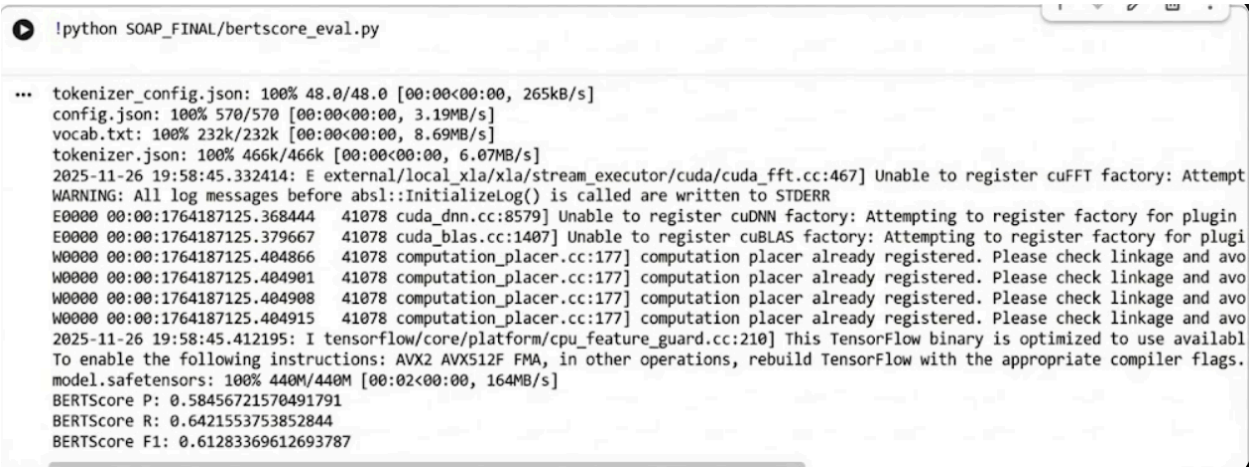
Example improvements:

- Baseline hallucination rate: high
- Fine-tuned + regex hallucination rate: notably lower
- Proper sectioning into Subjective, Objective, Assessment, and Plan became consistent

These findings confirm that lexical metrics alone underestimate model quality in clinical summarization.



6.1 Infer.py for single input (not batch input)



6.2 BERTscore on infer_test.py

7. Analysis & Conclusion

The project demonstrates that conversational clinical summarization requires more than scaling models. Data variety, evaluation choice, inference design, and

hallucination control substantially affect real-world usability. BERTScore aligned better with clinical correctness because it captures semantic similarity rather than surface-level wording. Regex-guided filtering proved surprisingly powerful as a lightweight post-training intervention.

Limitations include dataset size, lack of medical expert evaluation, and absence of grounding against knowledge bases. Still, results suggest the system could serve as a drafting assistant for clinicians, reducing documentation burden.

Future work may explore:

- Reinforcement learning from clinician feedback
- Retrieval-augmented reasoning
- Audio-to-SOAP end-to-end pipelines
- Hallucination detection classifiers
- Domain-specific decoding constraints

8. Demo / System

A working inference system accepts CSVs of transcripts and outputs corresponding structured SOAP summaries, enabling:

- Batch evaluation
- Plug-and-play integration
- Research reproducibility

The entire project is organized, documented, and reproducible from dataset download to final inference.

Future scope includes scalability, wherein each doctor could be able to send their patients these SOAP notes at the end of each working day.