Fine-Tuning NanoChat on MIMIC-IV Discharge Summaries

Project Report

Group 7a

Group members: *Yuxin Du, Saumya Roy*

May 5, 2026

Abstract

This project fine-tuned a NanoChat language model on discharge summaries from MIMIC-IV-Note. The aim was to adapt an ICU-oriented checkpoint to the style and content of clinical discharge documentation, then evaluate whether the resulting model better predicts held-out discharge-summary text and whether it can answer simple questions grounded in discharge-note contexts. The submitted model, `d20_for_icu_mimic_discharge_ft`, was continued from `d20_for_icu` step 700 and evaluated at step 1700. On the recreated held-out test split it achieved a language-modeling loss of 2.3646 and perplexity of 10.6402. A small synthetic QA-style evaluation produced Exact Match of 0.1400, token F1 of 0.1400, numeric MAE of 0.6250, classification accuracy of 0.1667, and a heuristic hallucination rate of 0.7400. These results show that fine-tuning improved next-token modeling of discharge summaries, but the model remains unreliable for direct clinical QA without stronger instruction tuning, better grounded evaluation data, and human clinical review.

1 Motivation and Problem Statement

Clinical discharge summaries contain condensed information about diagnoses, treatments, hospital course, medications, follow-up plans, and patient history. Because these notes are long, dense, and written in domain-specific language, adapting a language model to this text type may improve its ability to model and generate clinically styled text. The goal of this project was therefore to fine-tune an existing NanoChat checkpoint on MIMIC-IV discharge summaries and evaluate the resulting model on held-out discharge-note text.

The main research question was: *Does continued training on discharge summaries improve NanoChat's language-modeling performance on held-out discharge notes, and how reliable is the model on simple grounded QA-style probes?* We focused on two evaluation levels. First, we measured standard language-modeling loss and perplexity on validation and test splits. Second, we added task-oriented QA metrics—Exact Match, token F1, numeric MAE, classification accuracy, and a heuristic hallucination rate—to test whether generated answers were grounded in the provided clinical context.

2 Data

The data source was the MIMIC-IV-Note discharge-summary table, a deidentified clinical notes dataset released through PhysioNet [1]. The broader MIMIC-IV resource contains deidentified electronic health record data from Beth Israel Deaconess Medical Center [2]. We used the local file `discharge.csv` and selected the note text column `text`. Preprocessing was intentionally minimal: carriage returns and null characters were normalized, excessive whitespace was collapsed, clinical headings and line breaks were preserved, and notes outside the configured length range were excluded. In the final run, no rows were dropped by the length filters.

To reduce leakage, the split was performed by `subject_id` using a deterministic SHA-1 hash bucket. This ensured that all notes from the same patient appeared in only one split. Table 1 summarizes the resulting data. The split sanity checks found zero subject overlap between train, validation, and test. One duplicate note-text hash was detected within the dataset, but no duplicate text hashes crossed split boundaries.

Table 1: Dataset split statistics after preprocessing.

Split	Notes	Subjects	Admissions	Mean chars	Median chars	Actual ratio
Train	299,691	131,504	299,691	10,222.48	9,541	0.9032
Validation	16,521	7,344	16,521	10,259.59	9,613	0.0498
Test	15,581	7,066	15,581	10,208.45	9,511	0.0470
Total	331,793	–	331,793	–	–	1.0000

A token-length sanity check was also run on 200 sampled notes per split using the NanoChat tokenizer. Median token lengths were 2,957.5 for train, 2,646.0 for validation, and 2,829.0 for test. At least 97.5% of sampled notes in every split contained enough tokens for a 1,024-token training window, supporting the selected sequence length for fine-tuning and evaluation.

3 Model and Training Setup

The model is a transformer-based autoregressive language model following the general next-token prediction setup used by modern transformer language models [3]. Training continued from the existing NanoChat checkpoint `d20_for_icu` at step 700. The fine-tuned output checkpoint was saved as `d20_for_icu_mimic_discharge_ft`. The architecture loaded from the checkpoint metadata is shown in Table 2.

Table 2: Checkpoint architecture and evaluation configuration.

Parameter	Value
Checkpoint evaluated	`d20_for_icu_mimic_discharge_ft`, step 1700
Source checkpoint	`d20_for_icu`, step 700
Model sequence length in metadata	2048
Training/evaluation sequence length	1024
Vocabulary size	65,536
Layers	20
Attention heads	10
Key/value heads	10
Embedding dimension	1280
Loaded tensor dtype	bfloat16
Device used	CUDA

The main training hyperparameters are listed in Table 3. The objective was standard autoregressive next-token prediction on cleaned discharge-summary text. Validation loss and perplexity were evaluated every 100 fine-tuning steps. Checkpoints were saved in the NanoChat base-checkpoint layout so that the model could be opened through the web UI.

Table 3: Fine-tuning hyperparameters.

Hyperparameter	Value
Fine-tuning steps	1000
Source step $\rightarrow$ final step	700 $\rightarrow$ 1700
Batch size	2
Gradient accumulation	8
Learning rate	1×10^{-5}
Weight decay	0.01
Evaluation interval	every 100 steps
Evaluation batches	20
Save interval	every 500 steps
Random seed	1337
Train/validation/test target split	90% / 5% / 5% by subject
Minimum / maximum note length	200 / 200,000 characters

4 Evaluation Methods

Language-modeling evaluation used loss and perplexity on held-out notes. Perplexity was computed as exp(loss), so lower values indicate better prediction of held-out discharge-summary text. We reported the source checkpoint validation score, periodic validation scores during training, final validation loss, and final test loss.

To extend evaluation beyond next-token prediction, a QA-style test set was also generated from the recreated old held-out test split. The synthetic QA set contained 400 rows generated from 100 source notes: 100 exact-heading questions, 100 numeric questions about the number of occurrences of the word "discharge," and 200 yes/no classification questions about whether allergies or medications were mentioned. Due to generation cost, the QA generation run evaluated the first 50 examples. This QA set is useful for a controlled sanity check, but it is not equivalent to a manually curated clinical QA benchmark.

The added QA metrics were defined as follows:

- **Exact Match (EM):** normalized generated answer exactly equals the normalized reference answer.

- **Token F1:** token-overlap F1 between normalized prediction and reference.
- **Numeric MAE:** mean absolute error after extracting the first number from numeric predictions.
- **Classification accuracy:** accuracy on yes/no classification examples.
- **Heuristic hallucination rate:** rule-based unsupported-answer rate. It flags invalid answers, unsupported numeric/entity content, wrong classifications, and missing numeric outputs. This is only a proxy for hallucination and should be validated manually for clinical claims.

5 Results

Table 4 shows the main language-modeling results. The baseline ICU checkpoint had validation loss 3.5370 and validation perplexity 34.3635. The final evaluated checkpoint at step 1700 achieved final validation loss 2.1938 and final validation perplexity 8.9690, corresponding to a substantial improvement over the source model. On the recreated held-out test split, the model achieved loss 2.3646 and perplexity 10.6402. During periodic validation, the lowest observed validation loss was 2.0418 at step 1600, while the selected final submitted checkpoint was step 1700 because it was the final saved checkpoint used for the old-weights evaluation and web-UI smoke test.

Table 4: Language-modeling metrics.

Model / checkpoint	Step	Split	Notes	Loss	Perplexity
`d20_for_icu`	700	Validation	Source baseline	3.536995	34.363518
`d20_for_icu_mimic_discharge_ft`	1600	Validation	Best periodic validation	2.041837	7.704751
`d20_for_icu_mimic_discharge_ft`	1700	Validation	Final training validation row	2.083480	8.032374
`d20_for_icu_mimic_discharge_ft`	1700	Validation	Final re-evaluation	2.193774	8.969000
`d20_for_icu_mimic_discharge_ft`	1700	Test	Recreated old held-out test	2.364642	10.640230

Table 5 reports the QA-style metrics computed on 50 generated answers. The model performed poorly on direct grounded QA despite the improved language-modeling loss. Classification accuracy was only 0.1667 over 24 classification examples, and the heuristic hallucination proxy flagged 74% of the evaluated responses. This indicates that next-token fine-tuning alone did not make the model reliable for instruction-following or clinically grounded answer extraction.

Table 5: QA-style metrics from the old-weights evaluation notebook.

Metric group	Metric	Value	Evaluation count
Language modeling	Test loss	2.364642	20 batches
Language modeling	Test perplexity	10.640230	20 batches
QA evaluation	Exact Match	0.140000	50 examples
QA evaluation	Token F1	0.140000	50 examples
QA evaluation	Numeric MAE	0.625000	8 numeric examples
QA evaluation	Classification accuracy	0.166667	24 classification examples
QA evaluation	Heuristic hallucination rate	0.740000	50 examples

A breakdown by QA task type is shown in Table 6. Exact-heading questions had zero EM/F1 in the evaluated sample. Numeric questions performed better but still had a high unsupported-answer rate. Yes/no classification questions were the weakest group, with a hallucination proxy rate above 0.83.

Table 6: QA metric breakdown by task type.

Task type	n	Exact Match	Token F1	Hallucination proxy rate
Classification	24	0.166667	0.166667	0.833333
Exact heading	13	0.000000	0.000000	0.615385
Numeric	13	0.230769	0.230769	0.692308

6 Analysis and Discussion

The main positive result is that fine-tuning clearly improved discharge-summary language modeling. Compared with the source checkpoint, final validation perplexity decreased from 34.36 to 8.97 in the final re-evaluation, and

the recreated test perplexity was 10.64. This suggests that the model learned the distribution, formatting, and clinical vocabulary of discharge summaries better than the ICU source checkpoint.

However, the QA-style evaluation shows a gap between language modeling and reliable clinical question answering. The model often produced verbose answers to questions that expected short spans, yes/no labels, or numbers. The low classification accuracy suggests that the model was not consistently following the requested answer format. The hallucination proxy rate of 0.74 should not be interpreted as a clinically validated hallucination rate, but it is a strong warning sign that generated answers were frequently unsupported, malformed, or inconsistent with the reference rules.

The difference between the language-modeling results and QA results is expected because the training objective was next-token prediction, not supervised instruction following or extractive QA. A model can become better at predicting the next token in discharge summaries while still being poor at answering constrained questions. The current model is therefore more defensible as a domain-adapted language model than as a clinical assistant.

7 Limitations and Future Work

This study has several limitations. First, the QA benchmark was synthetic and small, with only 50 generated examples evaluated in the final run. Second, the hallucination metric was rule-based and should be considered a proxy rather than a clinical safety evaluation. Third, evaluation used only automatic metrics; a clinically meaningful assessment would require manual review by qualified clinicians. Fourth, the final submitted checkpoint was step 1700, although the best periodic validation loss appeared at step 1600. Future work should compare saved checkpoints more systematically, add early stopping, evaluate a manually curated QA set, and train the model with instruction-following examples if QA behavior is required.

A stronger final system would separate two goals: domain adaptation and safe clinical QA. Domain adaptation can be measured with loss and perplexity. Clinical QA should be evaluated with expert-reviewed questions, strict output formats, evidence citation from the source note, and manual error categories such as unsupported statement, wrong entity, wrong negation, and wrong temporal relation.

8 Conclusion

Fine-tuning NanoChat on MIMIC-IV discharge summaries substantially improved held-out language-modeling performance compared with the starting ICU checkpoint. The final evaluated model at step 1700 reached a recreated test loss of 2.3646 and perplexity of 10.6402. Nevertheless, the added QA metrics reveal that the model is not reliable for direct clinical QA: Exact Match and token F1 were both 0.14, classification accuracy was 0.1667, and the heuristic hallucination rate was 0.74. The best interpretation is that the project successfully produced a discharge-summary-adapted language model, but further supervised instruction tuning and clinically grounded evaluation are required before using it for question answering or decision support.

Acknowledgements

Add any human advisors, teaching assistants, or external contributors here. We also acknowledge the use of AI assistance for drafting and organizing parts of the report and notebooks; all experimental results, code, and final interpretations should be checked and approved by the group before submission.

References

[1] A. Johnson, T. Pollard, S. Horng, L. A. Celi, and R. Mark. *MIMIC-IV-Note: Deidentified free-text clinical notes*, version 2.2. PhysioNet, 2023. doi: 10.13026/1n74-ne17.

[2] A. E. W. Johnson, L. Bulgarelli, T. J. Pollard, S. Horng, L. A. Celi, and R. G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10, 1, 2023.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

A Submitted Code and Reproducibility Notes

The report was generated from two project notebooks:

- `mimic_v_nanochat_training_validation_webui_sanity_checks.ipynb`: preprocessing, split checks, fine-tuning, checkpoint saving, validation/test language-modeling metrics, and web-UI checkpoint smoke test.

- `mimic_discharge_hf_old_weights_test_eval.ipynb`: old-weight loading, old test-set recreation, language-modeling test evaluation, QA generation, Exact Match, token F1, numeric MAE, classification accuracy, and heuristic hallucination scoring.

Important generated or expected files:

- `clinical/mimic_discharge/train.jsonl`, `val.jsonl`, and `test.jsonl`
- `clinical/mimic_discharge/split_metadata.csv`
- `clinical/mimic_discharge/runs/d20_for_icu_mimic_discharge_ft_metrics.csv`
- `clinical/mimic_discharge/runs/d20_for_icu_mimic_discharge_ft_hf_test_metrics.csv`
- `clinical/mimic_discharge/runs/d20_for_icu_mimic_discharge_ft_qa_metrics.csv`
- `clinical/mimic_discharge/runs/d20_for_icu_mimic_discharge_ft_qa_summary.json`
- `$NANOCHAT_BASE_DIR/base_checkpoints/d20_for_icu_mimic_discharge_ft/model_001700.pt`
- `$NANOCHAT_BASE_DIR/base_checkpoints/d20_for_icu_mimic_discharge_ft/meta_001700.json`

Recommended environment and libraries include Python, PyTorch, pandas, matplotlib, tqdm, pyarrow, huggingface_hub, and the local NanoChat repository modules. The web interface can be started with:

```
python -m scripts.chat_web -g d20_for_icu_mimic_discharge_ft -s 1700 -i base
```