MELD: Memory-Enhanced Learning from Disagreements for Gradient-Free LLM Adaptation in Biomedical Text Classification

Anonymous ACL submission

Abstract

Large language models (LLMs) have shown promise for biomedical text classification, but adapting them to specialized tasks that require interpreting highly nuanced clinical narratives and and specialized medical knowledge remains challenging. Fine-tuning requires training infrastructure and produces model artifacts that are difficult to inspect, while few-shot prompting uses labeled examples only as temporary context. In this paper, we propose **MELD** (Memory-Enhanced Learning from Disagreements), a gradient-free supervised adaptation framework that converts model-label disagreements into persistent natural-language rules. During training, an LLM classifies each labeled example using a prompt that includes all rules accumulated so far. When a prediction is incorrect, a second LLM call is made to provide the gold-label correction and retries the example, preserving the full exchange as context, until the model predicts correctly or a maximum number of attempts is reached. After that, a third LLM call is made to extract a single generalizable rule from the resolved example and appended to the persistent memory. During inference, the full memory is inserted into the prompt directly, requiring no weight updates or retrieval index. We evaluated MELD on four biomedical NLP benchmarks using an institutional Gemma-family LLM sever configured with the model identifier `gemma-4`, with supervised RoBERTa as a reference baseline and GPT-5-mini as a secondary comparator. MELD outperformed the best zero/few-shot prompt baseline on every task and was competitive with RoBERTa on SMM4H-APO (0.89 vs. 0.88 full-data F_1) and Stigma-CN (0.71 vs. 0.73), although larger gaps remained on N2C2-MC and at the full-data SMM4H-MRC checkpoint. The secondary GPT-5-mini evaluation showed the same best-checkpoint pattern. Ablation experiments showed that removing the accumulated memory returned performance to approximately zero-shot levels. These results suggest that explicit memory construction can provide a practical and explainable mechanism for adapting LLMs to biomedical classification tasks.

1 Introduction

Biomedical text classification often requires task-specific knowledge that is not fully captured by standard large language model (LLM) prompts. Some of this knowledge is explicitly clinical, but much of it reflects implicit psycholinguistic reading strategies that humans apply without conscious awareness and that are difficult to pre-specify. For example, clinical notes may indicate a medication change through context around verbs such as "held" or "switched," whereas medication lists without such verbs are often indicative of no change. A similarly difficult challenge arises when clinical concepts must be mapped onto informal, heterogeneous real-world language, such as when social media users describe adverse pregnancy outcomes or medication-regimen changes without clinical terminology. Both challenges are difficult to anticipate and encode in a static prompt, yet both are central to text classification tasks such as medication-change detection, stigma detection, and social-media health monitoring (Mahajan et al., 2023; Magge et al., 2021; Harrigian et al., 2023).

The most common alternative, fine-tuning pre-trained models, has been an effective strategy for many clinical and health-related natural language processing (NLP) tasks. However, it requires substantial computational resources, and the resulting changes to model weights may be difficult to interpret, which is a critical limitation in clinical contexts where transparency and accountability matter (Hu et al., 2022; Doshi-Velez and Kim, 2017). Few-shot prompting offers a lighter-weight middle ground by placing labeled examples directly in the context window, but the model cannot learn from

its errors across interactions since no knowledge is retained between inputs. (Brown et al., 2020). Furthermore, research has shown that model performance in these approaches depends surprisingly little on whether the predicted labels during development are actually correct, suggesting that models exploit structural features such as format and label space rather than learning from the demonstrated input-label mappings themselves (Min et al., 2022). Thus, there remains a need for adaptation methods that leverage supervised labels without retraining the model and that produce interpretable, human-reviewable decisions.

We focus on text classification as our target task because it is widely recognized as a fundamental task in NLP and has been extensively studied and applied in biomedical and medical NLP settings, including medical text classification and biomedical literature classification.(Li et al., 2022; Wang et al., 2023). From clinical note analysis and diagnosis coding to the automated detection of adverse drug events, text classification serves as a critical enabler for clinical decision support systems and health information extraction (Yao et al., 2019; Mullenbach et al., 2018; Henriksson et al., 2015). By transforming unstructured clinical narratives into structured, actionable insights, it remains a central component of modern biomedical research and clinical practice.

We propose **MELD** (Memory-Enhanced Learning from Disagreements), a gradient-free, supervised adaptation framework for biomedical text classification, which learns an external natural-language memory from model-label disagreements. For each labeled training example, the LLM first generates a prediction conditioned on the current memory. When a prediction is incorrect, MELD provides the correct label, reruns the example, and prompts the model to extract a generalizable rule from the input and its gold-label classification. This rule is appended to a persistent memory file and applied to all subsequent training examples and held-out inference.

The resulting memory file is an auditable output: an append-only set of natural-language rules extracted from corrected errors, each traceable to a specific training disagreement. Unlike static knowledge bases or retrieval indices, these rules can be inspected, edited, or removed by domain experts without retraining the underlying model. This transparency is not simply a practical convenience but a meaningful property of clinical deployment contexts, where accountability requires that automated decisions are interpretable by and explainable to stakeholders.

We evaluated MELD on four biomedical NLP benchmarks, comparing it against zero-shot prompting, few-shot prompting, and supervised RoBERTa (Liu et al., 2019) using an institutional Gemma-family LLM sever (Gemma Team, 2024) as the primary model and GPT-5-mini (OpenAI, 2025) as a secondary. We also examined memory growth, duplicate rules, unmatched output rates, non-convergent corrections, API-call budgets, and ablations over memory, retry depth, and example ordering.

Our study makes the following contributions:

1. We developed a **gradient-free supervised adaptation framework** for biomedical text classification that converts corrected LLM mistakes into persistent natural-language rules.
2. We evaluated MELD with a **primary Gemma-family LLM sever** (Gemma Team, 2024) on four biomedical NLP benchmarks and compared it with zero-shot prompting, few-shot prompting, and supervised RoBERTa (Liu et al., 2019).
3. We conducted a **secondary GPT-5-mini analysis** (OpenAI, 2025) to assess whether the observed results are specific to a single model.
4. We performed **mechanistic and operational analyses** of memory ablation, correction-attempt depth, example ordering, rule growth, API cost, unmatched outputs, and non-convergent corrections.

2 Related Work

Iterative self-correction. Madaan et al. (2023) propose Self-Refine, where an LLM generates, critiques, and revises a single output iteratively. Shinn et al. (2023) introduce Reflexion, which stores natural-language reflections on past task failures in episodic memory for agentic planning. MELD is related to these methods because it also uses model feedback after an initial error. However, Self-Refine focuses on revising a single output, and Reflexion stores task trajectories for agentic planning. In contrast, MELD uses supervised labels to create cross-example rules for classification, and these rules persist across the full training run.

Retrieval-augmented generation. RAG (Lewis et al., 2020) and Atlas (Izacard et al., 2023) in-

ject passages retrieved from a static corpus. Like MELD, these methods augment the prompt at inference time. The main difference is that their retrieved knowledge is pre-existing and factual, whereas MELD constructs task-specific rules from corrected training errors.

Memory-augmented systems. Two recent preprints describe related memory-augmented LLM systems. Li (2025) introduce MALM, a unified architecture that appends a single trainable memory-attention layer to a frozen base LLM for knowledge retrieval and safety compliance. Dong et al. (2026) present MALMAS, a multi-agent system with three memory types (procedural, feedback, and conceptual) for automated feature generation on tabular data. Both works share the broad goal of equipping LLMs with persistent memory but differ from MELD in several important ways. MALM encodes memory as latent attention weights, i.e., sub-symbolic vectors inaccessible to human inspection, and still requires gradient-based training of the memory layer. MALMAS is designed for tabular feature engineering via multi-agent routing, not text classification. In contrast, MELD stores memory as plain-text rules, injects the full memory into the system prompt, requires zero gradient steps, and targets supervised biomedical text classification.

3 MELD: Memory-Enhanced Learning from Disagreements

3.1 Problem Formulation and Task Configuration

We consider supervised binary classification with labels $y_i \in \{0, 1\}$. Given an ordered training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ and a frozen LLM $\mathcal{L}$, MELD builds a memory M of natural-language rules. At prediction time, M is inserted into the system prompt; the model weights are never updated. For each task, the implementation requires a configuration τ containing: (i) a task system prompt, (ii) a user-prompt template for inserting the input text, (iii) positive and negative indicator phrase lists used to parse model outputs, (iv) one correction message for each gold label, and (v) a separate rule-extraction system prompt and user template. Appendix J reports the exact prompt templates, label indicators, and correction messages used in the experiments. Figure 1 illustrates the full framework.

3.2 Training: Iterative Memory Construction

Algorithm 1 summarizes the memory construction procedure. Training processes the first N rows of the training CSV in file order. Before each example, the current memory file is loaded and appended to the task system prompt under the heading **Learned Rules**. The classification prompt uses the full input text. If the first prediction is correct, MELD records no new rule. If the prediction is incorrect or cannot be parsed, the model response is kept in the conversation history, a gold-label correction message is appended, and the same example is retried until it is correct or reaches K total classification attempts.

When a later attempt eventually produces the correct parsed label, MELD makes one additional *rule-extraction call*. This call receives the task-specific extraction system prompt and a user message containing the first 800 characters of the input text plus the verbalized correct classification. The extraction prompt asks for one concise, generalizable sentence and forbids patient- or user-specific details. The extracted sentence is appended to M and is available starting with the next training example.

MELD extracts rules only from corrected errors. Correct first-attempt predictions do not add new rules, and examples that never produce the correct parsed label within K classification attempts are excluded from rule extraction. Because at most one rule is added per corrected example, the memory size is bounded by the number of converged corrections and therefore by N. During training, MELD does not deduplicate, merge, rewrite, or delete rules.

3.3 Memory Format

Memory is stored as a plain Markdown file with a flat list of dated bullet points:

```
- [date]   "held" without restart => change
- [date]   medication list only => no change
```

During training, the implementation maintains a live memory and, at predefined training sizes, writes immutable memory snapshots. At inference, a selected snapshot is appended verbatim to the system prompt under a **Learned Rules** section. Single-sentence rules keep the memory relatively compact, and no retrieval index is used. Because the full memory is injected verbatim, prompt length grows with the number of accumulated rules.

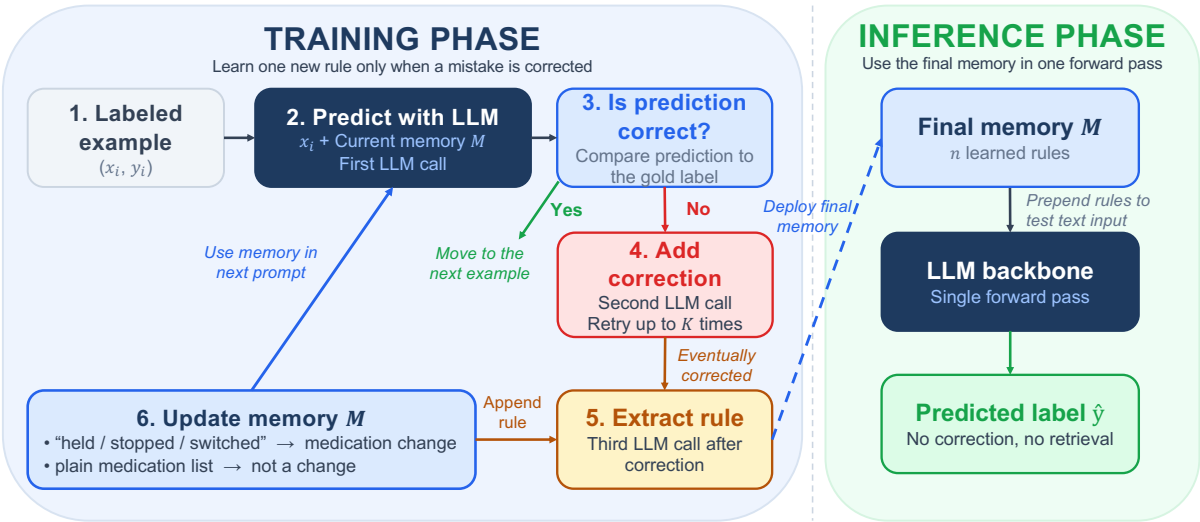


Figure 1: The MELD framework, demonstrated using the medication change classification task example. **Training phase (left):** For each labeled example (x_i, y_i), the LLM predicts using the current memory M injected into the system prompt. If wrong, a correction message is appended and the model is queried for up to K total classification attempts; once corrected, a second LLM call extracts one generalizable rule that is appended to M. The updated memory is fed back into the next training example via the system prompt. **Inference phase (right):** The final memory is appended verbatim to the task system prompt; each test example is classified in a single forward pass with no correction or retrieval.

Algorithm 1 MELD Training Algorithm

Require: ordered $\mathcal{D}$, LLM $\mathcal{L}$, task config τ, max attempts K
$M \leftarrow \emptyset$
for each $(x_i, y_i) \in \mathcal{D}$ **do**
  sys $\leftarrow \tau$.system $\|$ FormatMem(M)
  msgs $\leftarrow [(\texttt{user}, \tau.\text{user_template}(x_i))]$
  for $a = 1, \ldots, K$ **do**
    $r \leftarrow \mathcal{L}(\text{sys, msgs; 256 tokens})$
    $\hat{y} \leftarrow$ Match(r, τ)
    if $\hat{y} = y_i$ **then**
      if $a > 1$ **then** ▷ corrected after at least one error
        $v \leftarrow$ Verbalize(y_i, τ)
        $x_i' \leftarrow$ Trunc$_{800}(x_i)$
        $u \leftarrow \tau$.extract_template(x_i', v)
        $\rho \leftarrow \mathcal{L}(\tau.\text{extract_system}, u; \text{256 tokens})$
        $M \mathrel{+}= $ DatedBullet(ρ)
      end if
      break
    else
      append $(\texttt{assistant}, r)$ to msgs
      append $(\texttt{user}, \tau.\text{correction}(y_i))$ to msgs
    end if
  end for
end for
return M

3.4 Inference and Label Matching

Task-specific *indicator* phrase lists parse LLM outputs into binary labels. Longer phrases are matched first to prevent partial shadowing (e.g., "not discussed" before "discussed"). At inference, the selected memory snapshot is injected into the system prompt and each test example is classified with a single model call and no correction step. Unmatched responses are logged as parser failures and mapped to the negative class only for metric computation. Thus, inference has the same one-call-per-example structure as zero-shot prompting, but with a longer system prompt.

4 Experiments

Datasets. We evaluate on four biomedical NLP benchmarks spanning clinical EHR notes and social media: **(1) N2C2-MC** (n2c2 2022): a binary medication-change detection task derived from the 2022 contextualized medication event extraction challenge in clinical notes (Mahajan et al., 2023); **(2) SMM4H-MRC** (SMM4H 2021 Task 3a): self-reported changes in medication treatment on social media, with 91.23% negative examples in the full training split and 91.16% negative examples in the test split (Magge et al., 2021); **(3) Stigma-CN**: stigmatizing language toward patients in clinical notes (Harrigian et al., 2023); **(4) SMM4H-APO** (SMM4H 2021): adverse pregnancy outcome tweets (Magge et al., 2021).

Backbones and implementation. The primary experiments use an institutional OpenAI-compatible Gemma-family LLM sever (Gemma Team, 2024) with model identifier `gemma-4`; we refer to this LLM sever as **Gemma-4** through-

out the results. We additionally evaluate GPT-5-mini (OpenAI, 2025) with the same zero-shot, few-shot, and MELD checkpoint protocol as a secondary backbone analysis; the ablations and supervised RoBERTa comparison (Liu et al., 2019) remain centered on Gemma-4. Classification and rule-extraction calls both use max 256 tokens. Default hyperparameters are $K=10$ classification attempts and training sizes $n \in \{50, 100, 200, 500, 1{,}000, 2{,}000, 5{,}000\}$ plus the full training size when available. All saved checkpoints at $n \geq 50$ are evaluated. For Stigma-CN, the largest checkpoint is the full-data $n=799$ prefix.

Baselines. *Zero-shot*: task system prompt, no examples or memory. *Few-shot* ($k \in \{1, 4, 8, 16, 32\}$): standard in-context learning with k fixed-seed, class-balanced training examples inserted into the system prompt; demonstration inputs are truncated to 200 characters. *RoBERTa (supervised)*: `roberta-base` fine-tuned at the same learning-curve checkpoints as MELD, serving as a supervised reference (Liu et al., 2019). Appendix G gives the training details. We do not claim head-to-head superiority over batch instruction-induction methods such as Instruction Induction (Honovich et al., 2023), APE (Zhou et al., 2023), or APO (Pryzant et al., 2023); those approaches require a separate prompt-search loop and typically a development-set selection step, whereas MELD studies online rule accumulation under a fixed task prompt. Performance is measured by F_1 on the positive class, with uncertainty estimated using 95% bootstrap confidence intervals (1,000 resamples).

5 Results

5.1 Comparative Performance

Table 1 summarizes the primary Gemma-4 comparison. Figure 2 shows 95% bootstrap confidence intervals for the MELD and RoBERTa learning curves, so small table-only differences should be interpreted conservatively. Appendix A reports the full checkpoint grid.

Comparison with prompting baselines. The results show that, using Gemma-4, the best MELD checkpoint exceeds the best zero/few-shot prompt baseline on all four tasks. Even with $n=50$ examples, MELD matches or exceeds the best few-shot configuration on 2 of 4 tasks. The largest gains relative to the strongest zero/few-shot prompt baseline are on N2C2-MC (+7.40 percentage points at $n=5{,}000$), SMM4H-MRC (+7.52 percentage points at $n=1{,}000$), and SMM4H-APO (+4.40 percentage points at $n=1{,}000$).

Comparison with supervised RoBERTa. Compared with supervised RoBERTa, MELD is most competitive on SMM4H-APO and Stigma-CN. On SMM4H-APO, MELD reaches $F_1 = 0.89$ at $n=1{,}000$, effectively matching full-data RoBERTa (0.88 at $n=5{,}189$) and remaining within 0.50 percentage points of RoBERTa's best checkpoint (0.89 at $n=5{,}000$). On Stigma-CN, MELD reaches $F_1 = 0.71$ at the full-data $n=799$ checkpoint, compared with full-data RoBERTa at 0.73.

The performance gap is larger on N2C2-MC and SMM4H-MRC. On N2C2-MC, the best MELD checkpoint reaches 0.76 at $n=5{,}000$ and 0.75 at the full-data $n=5{,}247$ checkpoint, compared with RoBERTa at 0.87 and 0.87 at the corresponding checkpoints. This gap is consistent with the task's highly implicit clinical syntax. On SMM4H-MRC, both approaches are limited by class imbalance: MELD reaches 0.56 at $n=1{,}000$ and 0.49 at full data, while RoBERTa reaches 0.57 at full data and peaks at 0.62 at $n=5{,}000$. The RoBERTa curve also shows a low-data majority-class failure mode on this task, consistent with the sparse positive class.

Late and full-data checkpoints. Adding more training examples has task-dependent effects rather than a uniformly positive trend. Best MELD performance occurs before the full-data checkpoint on three tasks, which suggests that append-only memory benefits from future work on rule filtering, scheduling, or compression. The full checkpoint values are reported in Appendix A.

5.2 Learning Dynamics

Figure 2 presents F_1 vs. training size n across all saved checkpoints for MELD and RoBERTa, with zero-shot and best-few-shot as horizontal reference lines. The curves show that most gains from MELD occur early. The steepest improvement is typically between $n=0$ and $n=100$, suggesting that the memory captures many high-impact heuristics within the 100 samples. After this early phase, the trajectories become task-dependent: SMM4H-APO improves steadily up to $n=1{,}000$ and remains stable through later checkpoints; N2C2-MC dips at intermediate checkpoints before reaching its highest observed F_1 at $n=5{,}000$; SMM4H-MRC

Table 1: Main Gemma-4 results. F_1 for the positive class with 95% bootstrap confidence intervals in brackets. Best few-shot and best MELD report the strongest checkpoint within each block; full-data MELD reports the largest available training prefix. RoBERTa is a supervised reference baseline.

Method	N2C2-MC	SMM4H-MRC	Stigma-CN	SMM4H-APO
Best MELD	0.76 [0.72,0.80]	0.56 [0.49,0.62]	0.71 [0.64,0.78]	**0.89** [0.87,0.91]
MELD full-data	0.75 [0.71,0.79]	0.49 [0.43,0.55]	0.71 [0.64,0.78]	0.88 [0.86,0.90]
RoBERTa full-data	**0.87** [0.83,0.90]	**0.57** [0.49,0.64]	**0.73** [0.65,0.80]	0.88 [0.86,0.90]
Zero-shot	0.64 [0.60,0.68]	0.41 [0.33,0.47]	0.67 [0.57,0.75]	0.75 [0.72,0.77]
Best few-shot	0.69 [0.64,0.72]	0.48 [0.42,0.55]	0.67 [0.59,0.75]	0.84 [0.82,0.86]

rises through n=1,000 before falling at later checkpoints; and Stigma-CN improves through n=200, softens at n=500, then recovers to 0.71 at the full-data n=799 run.

The SMM4H-MRC curve also shows that class imbalance affects fine-tuning and memory learning differently. RoBERTa collapses to a majority-class solution at small n and begins to recover only at n=1,000 and later checkpoints. Appendix C reports the full positive/negative distribution for each plotted checkpoint in Table 4. For the smaller gaps between mature checkpoints, we interpret the curves conservatively and avoid strong significance claims based only on Table 1.

Secondary backbone check. To assess whether the pattern is specific to a single LLM sever, we repeated the zero-shot, few-shot, and MELD checkpoint protocol with GPT-5-mini. At the best saved checkpoint, GPT-5-mini MELD matches or exceeds the strongest GPT-5-mini prompt baseline on all four tasks, although the gains are again non-monotonic for some late checkpoints. Appendix B reports the secondary table and learning curves.

5.3 Memory Growth, Cost, and Failure Modes

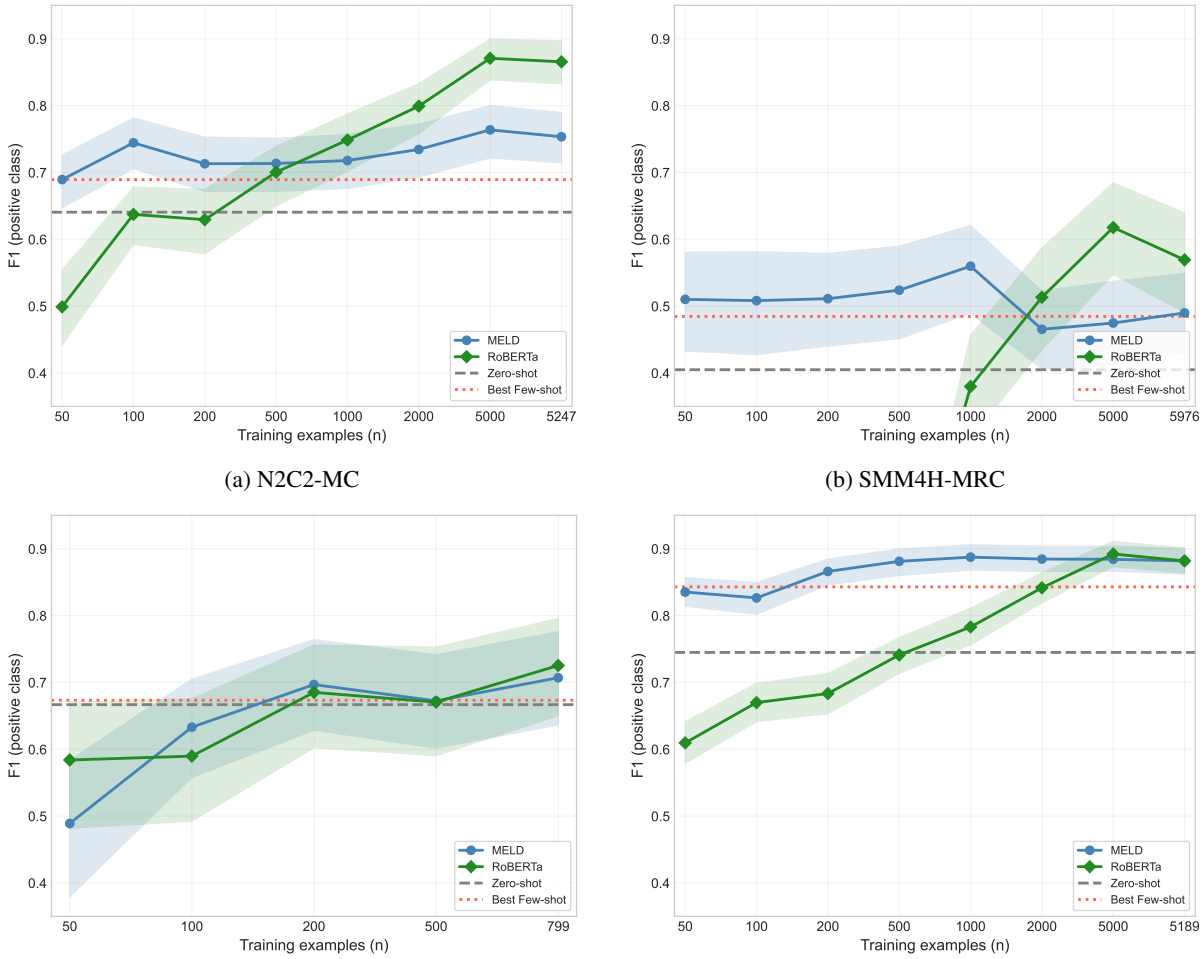
MELD's memory grows with the number of corrected errors because each corrected error contributes at most one new rule. In practice, rule growth is substantially sublinear across tasks, and average rule length remains stable at roughly 23–30 words per rule (Appendix E). The full-data runs also show that performance is not determined by memory size alone: SMM4H-APO remains stable after n=1,000 despite continued rule growth, whereas SMM4H-MRC accumulates the largest memory but remains limited by precision. Label-parser failures are not the explanation for this pattern; unmatched outputs are zero at the reported full-data checkpoints. Training cost is instead governed by the correction budget: with n examples and maximum depth K, MELD makes between n and nK classification calls, plus at most n rule-extraction calls, while inference remains one call per test example.

6 Rule Quality and Interpretability

The examples show three properties that make MELD's memory useful for expert review. First, they are *generalizable*: framed as universally applicable heuristics, not instance-specific observations. Second, they are *contrastive*: the model articulates both the positive condition and the confusable negative. Third, they are *inspectable*: a clinical informatics team can inspect, correct, or delete individual rules without retraining. This audit capability is unavailable in standard fine-tuned models, although it should not be interpreted as external clinical validation of the rules.

7 Discussion

When MELD works best. The results suggest that MELD performs best when the positive class has coherent and articulable features, the class distribution is not extremely imbalanced, and the decision boundary can be expressed as natural-language heuristics. SMM4H-APO best reflects this setting. In contrast, SMM4H-MRC remains difficult because positive examples are sparse, the final memory is recall-heavy, and precision becomes the main bottleneck (Appendix D). The RoBERTa curve provides a related observation. With only 5–38 positive examples in the first 50–500 training instances, standard fine-tuning can converge to an all-negative or nearly all-negative classifier. MELD avoids this exact behavior because the task prompt and correction loop keep the positive label available, but its low precision shows that memory induction alone does not solve severe positive sparsity. Future work on such tasks should consider class-balanced sampling, loss reweighting, threshold tuning, and memory scheduling.



(a) N2C2-MC

(b) SMM4H-MRC

(c) Stigma-CN

(d) SMM4H-APO

Figure 2: F_1 learning curves across all comparable saved checkpoints at $n \geq 50$ on a log-scaled x-axis and full-size checkpoints when available. Shaded regions show 95% bootstrap confidence intervals. The blue curve denotes MELD; the green diamond curve denotes RoBERTa fine-tuning. Horizontal reference lines denote zero-shot (gray dashed) and best few-shot (red dotted).

Memory size. Because MELD injects the full memory, scalability is governed directly by rule count and prompt length (Section 5.3). The append-only design makes this tradeoff easy to measure, but it leaves redundant rules, conflicting rules, malformed rules, and rule revision unresolved. The supplementary ordering sweep further suggests that rule sequence can shift the final precision–recall operating point. Future work should explore memory compression, contradiction resolution, selective retrieval, and rule scheduling without changing the core correction loop.

Practical deployment. MELD requires only API access; no GPU is needed during training. For clinical teams without machine learning infrastructure, this framework can be a practical advantage. However, the approach is not cost free: training cost scales with correction attempts and can reach thousands of API calls at large n. The supplementary ablations show that this cost-quality tradeoff is task-dependent: smaller attempt budgets can reduce call volume and sometimes improve F_1, but the best setting is not identical across tasks. Compared with annotation-efficient fine-tuning, MELD trades inference latency (one API call per test example) for zero training infrastructure and an inspectable memory. This tradeoff may be useful when deployment simplicity and auditability are important.

While our evaluation focused on binary classification tasks, the MELD framework is inherently adaptable to multi-label classification scenarios, where input text may be associated with multiple target categories simultaneously. Furthermore, a key advantage of this framework is its utility for medical and other domain-specific experts who

Learned Rules: N2C2 Medication Change

- When a note mentions a medication "held" without restart instructions, it indicates a change.
- Lists of medications without action verbs (start / stop / increase) are not changes.
- "Started on" or "switched to" describes a change even in passive voice.
- "Continued at same dose" is not a change despite mentioning the drug.

Learned Rules: Adverse Pregnancy Outcomes

- NICU admission or premature birth in a tweet signals an adverse outcome.
- Positive anticipation about an ongoing pregnancy without complications is not an adverse outcome.
- First-person use of "loss" or "miscarriage" indicates an adverse outcome even without medical terminology.
- A pregnancy described in past tense with negative emotional language likely describes an adverse outcome.

Figure 3: Representative rules from trained MELD memories. Rules are generalizable, domain-faithful, contrastive, and concise, with one sentence each.

lack extensive machine learning engineering backgrounds. By providing a gradient-free, intuitive interface that converts disagreements into interpretable natural-language rules, MELD empowers these stakeholders to transcend the limitations of basic prompting, allowing them to develop, inspect, and refine highly interpretable models tailored to their specific classification needs without the necessity for complex retraining infrastructure.

8 Conclusion

In this work, we developed MELD, a gradient-free supervised adaptation framework that converts model-label disagreements into a persistent memory of natural-language rules. Across four biomedical NLP benchmarks, MELD consistently improved over zero/few-shot prompting in the primary Gemma-4 experiments without updating model weights. MELD was competitive with supervised RoBERTa on SMM4H-APO and Stigma-CN, although larger gaps remained on N2C2-MC and on full-data SMM4H-MRC. The secondary GPT-5-mini analysis showed that MELD matched or exceeded the strongest GPT-5-mini prompt baseline at the best saved checkpoint on all four tasks. The ablation experiments showed that the accumulated memory is the main source of performance improvement. Overall, these findings suggest that explicit memory construction is a promising strategy for adapting LLMs to biomedical text classification tasks, especially in settings where interpretability, auditability, and limited training infrastructure are important. Future work should evaluate MELD on additional backbones and clinical tasks, develop memory compression and rule-filtering methods, and assess how closely induced rules align with expert clinical guidance.

9 Limitations

MELD remains a sequential training procedure, and parallelizing rule induction would require conflict resolution for simultaneously extracted rules. The primary claims rely on Gemma-4, with GPT-5-mini reported as a secondary replication; this broadens the evidence but does not establish cross-family robustness. Rule quality has not been validated against expert clinical guidelines; thus, the current interpretability claim is structural rather than clinical. We also do not include direct comparisons to batch instruction-induction baselines such as Instruction Induction (Honovich et al., 2023), APE (Zhou et al., 2023), or APO (Pryzant et al., 2023). Finally, the append-only memory allows duplicate and malformed rules to persist, and SMM4H-MRC shows that class imbalance can still produce difficult precision–recall tradeoffs even when unmatched responses are eliminated. Cross-family validation, expert review of induced rules, and automated deduplication / rule-filtering mechanisms therefore remain necessary next steps.

10 Ethical Considerations

This work uses biomedical and health-related text classification benchmarks, including clinical notes and social media posts. Such data can contain sensitive health information, even when distributed through controlled benchmark releases or after de-identification. We therefore treat MELD as a method for supervised model adaptation and analysis rather than as a deployable clinical decision system. We did not collect new human-subject data; all experiments use existing benchmark datasets distributed through their respective de-identification, credentialing, or shared-task access processes. The paper reports aggregate quantitative results and representative generalized rules, and does not reproduce patient notes, user posts, or other potentially identifying examples. We also checked the manuscript and representative rules

to ensure that they do not contain names, handles, record identifiers, verbatim raw examples, or other text that could uniquely identify individual patients or social media users.

MELD's memory mechanism creates an additional privacy consideration: rules are induced from individual training disagreements and are stored persistently. If rule extraction were allowed to copy instance-specific details, a memory file could inadvertently retain sensitive information from the training data. To mitigate this risk, the rule-extraction prompts explicitly instruct the model to avoid patient- or user-specific details and to produce one generalizable rule. Any real deployment should additionally subject memory files to privacy review, de-identification checks, access control, and domain-expert auditing before sharing or reuse.

Errors in biomedical classification can have asymmetric harms. False negatives may miss medication changes, adverse pregnancy outcomes, or stigmatizing language that should be reviewed, while false positives may create unnecessary review burden or mischaracterize patient narratives. Because our experiments show task-dependent precision–recall tradeoffs, MELD should not be used as an autonomous decision maker in clinical or public-health workflows. Appropriate use would require human oversight, calibration to the downstream risk tolerance, and evaluation on the target population and institution before deployment.

Finally, induced rules may reflect biases present in the training annotations, benchmark construction, clinical documentation practices, or social media data. The interpretability of natural-language memory helps expose such rules for inspection, but it does not guarantee that they are clinically valid, fair, or robust across demographic groups and domains. Future use of MELD in sensitive settings should include subgroup evaluation where feasible, expert review of learned rules, and mechanisms for removing or revising rules that encode spurious, stigmatizing, or otherwise harmful patterns.

Code Availability

The MELD implementation and synthetic demo will be made available at (*redacted to maintain anonymity*).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Fengxian Dong, Zhi Zheng, Xiao Han, Wei Chen, Jingqing Ruan, Tong Xu, Yong Chen, and Enhong Chen. 2026. Memory-augmented llm-based multi-agent system for automated feature generation on tabular data. *arXiv preprint arXiv:2604.20261*.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Gemma Team. 2024. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Keith Harrigian, Ayah Zirikly, Brant Chee, Alya Ahmad, Anne Links, Somnath Saha, Mary Catherine Beach, and Mark Dredze. 2023. Characterization of stigmatizing language in medical records. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–329. Association for Computational Linguistics.

Aron Henriksson, Maria Kvist, Hercules Dalianis, and Martin Duneld. 2015. Identifying adverse drug event information in clinical notes with distributional semantic representations of context. *Journal of Biomedical Informatics*, 57:333–349.

Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. 2023. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1935–1952. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim

Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Jiarui Li. 2025. Memory-augmented language models (MALM): A unified architecture for knowledge integration and safety compliance. *Authorea Preprints*.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2):1–41.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36.

Arjun Magge, Ari Klein, Antonio Miranda-Escalada, Mohammed Ali Al-Garadi, Ilseyar Alimova, Zulfat Miftahutdinov, Eulalia Farre, Salvador Lima López, Ivan Flores, Karen O'Connor, Davy Weissenbacher, Elena Tutubalina, Abeed Sarker, Juan Banda, Martin Krallinger, and Graciela Gonzalez-Hernandez. 2021. Overview of the sixth social media mining for health applications (#SMM4H) shared tasks at NAACL 2021. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 21–32. Association for Computational Linguistics.

Diwakar Mahajan, Jennifer J Liang, Ching-Huei Tsou, and Ozlem Uzuner. 2023. Overview of the 2022 n2c2 shared task on contextualized medication event extraction in clinical notes. *Journal of Biomedical Informatics*, 144:104432.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064. Association for Computational Linguistics.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1101–1111.

OpenAI. 2025. Introducing GPT-5 for developers. OpenAI API Platform Blog.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36.

Yu Wang, Yuan Wang, Zhenwan Peng, Feifan Zhang, Luyao Zhou, and Fei Yang. 2023. Medical text classification based on the discriminative pre-training model and prompt-tuning. *Digital Health*, 9:20552076231193213.

Li Yao, Chengsheng Mao, and Yuan Luo. 2019. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC Medical Informatics and Decision Making*, 19(Suppl 3):71.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. In *Proceedings of the 11th International Conference on Learning Representations*.

A Full Gemma-4 Results

Table 2 reports the full point-estimate grid for the primary Gemma-4 experiments.

B Secondary GPT-5-mini Results

To assess backbone sensitivity, we repeated the zero-shot, few-shot, and MELD learning-curve experiments with GPT-5-mini. We treat these results as secondary because the main ablations and RoBERTa comparison use Gemma-4. At the best saved checkpoint, GPT-5-mini MELD matches or exceeds the strongest GPT-5-mini prompt baseline on all four tasks. The largest gains are on SMM4H-APO (0.86 vs. 0.75) and Stigma-CN (0.65 vs. 0.60). N2C2-MC improves at n=1,000 (0.72 vs. 0.65 zero-shot and 0.64 best few-shot) but falls to 0.64 at full data, so additional examples do not monotonically help. SMM4H-MRC is essentially tied with the best few-shot baseline (0.49 vs. 0.49) and remains difficult because of positive sparsity.

The full-data GPT-5-mini runs produce zero unmatched outputs on all four tasks, so their late-stage behavior cannot be attributed to label-parser failures. Earlier checkpoints still have unmatched outputs, reinforcing that output stability depends on both the backbone and the checkpoint.

C Data Statistics

Table 4 reports the class distribution for each training prefix used in Figure 2, along with the corresponding fixed test-set distribution. Training counts are reconstructed from the saved MELD training logs, which preserve the original sequential example order.

D Confusion Matrices

Figure 5 shows confusion matrices at the full training-size MELD checkpoint for each task. N2C2-MC and SMM4H-APO exhibit high recall (few false negatives), appropriate for patient-safety applications where missed events are costly. Stigma-CN also achieves high recall (0.86) at the cost of precision (0.60), consistent with a wide-net strategy for detecting harmful language. SMM4H-MRC remains the task with the clearest precision bottleneck, attributable in part to the 91.16% negative class distribution in the test split.

N2C2-MC (n=5,247)**.** With TP = 223, FP = 104, and FN = 42, MELD achieves high recall (0.84) at the cost of moderate precision (0.68). For medication-safety monitoring, this error profile favors missed-event reduction over false-alarm minimization.

SMM4H-MRC (n=5,976)**.** With TP = 96, FP = 164, and FN = 36, recall is relatively high (0.73) but precision is low (0.37). The task's severe class imbalance makes false positives the main error mode in the updated full-data run.

Stigma-CN (n=799)**.** The model shows high recall (0.86, FN = 12) but lower precision (0.60, FP = 51), indicating that the learned rules cast a wide net for stigmatizing language and flag many borderline negative cases.

SMM4H-APO (n=5,189)**.** This task shows the most balanced performance: precision 0.86, recall 0.90, with FP = 86 and FN = 56 out of 1,298 test examples.

E Memory Statistics

Table 5 summarizes exact memory-growth and failure-mode statistics from the saved logs. We additionally report the latest full-data checkpoints for all four tasks.

Exact redundancy is modest at standardized checkpoints but grows noticeably at full data. After stripping timestamps, the latest N2C2-MC memory contains 70 excess exact duplicate lines out of 701 rules, SMM4H-MRC contains 10 out of 971, SMM4H-APO contains 39 out of 549, and Stigma-CN contains 6 out of 240. These observations suggest that lightweight deduplication and rule-quality filtering may be more appropriate than aggressive truncation alone.

At the largest available checkpoints, non-converged training examples are rare: 0.44% on N2C2-MC, 0.30% on SMM4H-MRC, 0.40% on SMM4H-APO, and 0.38% on Stigma-CN. On held-out evaluation, unmatched outputs are zero for all four tasks at the reported checkpoints in the updated evaluation. Thus the persistent SMM4H-MRC difficulty is not attributable to label-parser failures; it instead reflects a task-specific precision–recall tradeoff. In the latest full-data runs, the realized training budgets remain below the theoretical worst-case bound: 6,856 total API calls for N2C2-MC, 8,080 for SMM4H-MRC, 1,306 for Stigma-CN, and 6,476 for SMM4H-APO.

Table 2: Full Gemma-4 results. F_1 for the positive class with 95% bootstrap confidence intervals in brackets. **Bold**: best result within the zero/few-shot block or within the MELD block for each task. The full-data MELD row uses task-specific training sizes: n=5,247, 5,976, 799, and 5,189, respectively. RoBERTa is a supervised full-data reference baseline.

Method	N2C2-MC	SMM4H-MRC	Stigma-CN	SMM4H-APO
RoBERTa (full-data)	0.87 [0.83,0.90]	0.57 [0.49,0.64]	0.73 [0.65,0.80]	0.88 [0.86,0.90]
Zero-shot	0.64 [0.60,0.68]	0.41 [0.33,0.47]	0.67 [0.57,0.75]	0.75 [0.72,0.77]
Few-shot k=1	0.58 [0.53,0.62]	0.46 [0.39,0.52]	0.66 [0.58,0.73]	0.81 [0.79,0.83]
Few-shot k=4	0.61 [0.56,0.65]	0.48 [0.41,0.54]	**0.67** [0.59,0.75]	0.82 [0.80,0.84]
Few-shot k=8	0.68 [0.64,0.71]	**0.48** [0.42,0.55]	0.67 [0.59,0.74]	0.82 [0.80,0.84]
Few-shot k=16	0.65 [0.60,0.69]	0.43 [0.37,0.48]	0.59 [0.48,0.67]	**0.84** [0.82,0.86]
Few-shot k=32	**0.69** [0.64,0.72]	0.37 [0.32,0.42]	0.66 [0.57,0.74]	0.84 [0.82,0.86]
MELD n=50	0.69 [0.65,0.73]	0.51 [0.43,0.58]	0.49 [0.38,0.58]	0.84 [0.81,0.86]
MELD n=100	0.74 [0.71,0.78]	0.51 [0.43,0.58]	0.63 [0.56,0.70]	0.83 [0.80,0.85]
MELD n=200	0.71 [0.67,0.75]	0.51 [0.44,0.58]	0.70 [0.63,0.76]	0.87 [0.85,0.88]
MELD n=500	0.71 [0.67,0.75]	0.52 [0.45,0.59]	0.67 [0.60,0.74]	0.88 [0.86,0.90]
MELD n=1,000	0.72 [0.68,0.76]	**0.56** [0.49,0.62]	—	**0.89** [0.87,0.91]
MELD n=2,000	0.73 [0.69,0.77]	0.47 [0.41,0.52]	—	0.88 [0.87,0.90]
MELD n=5,000	**0.76** [0.72,0.80]	0.47 [0.41,0.54]	—	0.88 [0.87,0.90]
MELD full-data	0.75 [0.71,0.79]	0.49 [0.43,0.55]	**0.71** [0.64,0.78]	0.88 [0.86,0.90]

Table 3: Secondary GPT-5-mini results. F_1 for the positive class with 95% bootstrap confidence intervals in brackets. Best few-shot and best MELD report the highest checkpoint within that block; full-data MELD reports the largest available training prefix.

Method	N2C2-MC	SMM4H-MRC	Stigma-CN	SMM4H-APO
Zero-shot	0.65 [0.60,0.69]	0.40 [0.33,0.47]	0.50 [0.39,0.60]	0.70 [0.67,0.73]
Best few-shot	0.64 [0.60,0.68] (k=32)	0.49 [0.41,0.56] (k=4)	0.60 [0.50,0.68] (k=1)	0.75 [0.72,0.78] (k=16)
Best MELD	0.72 [0.67,0.75] (n=1,000)	0.49 [0.42,0.55] (n=5,976)	0.65 [0.58,0.72] (n=50)	0.86 [0.83,0.88] (n=5,189)
Full-data MELD	0.64 [0.60,0.68] (n=5,247)	0.49 [0.42,0.55] (n=5,976)	0.65 [0.57,0.71] (n=799)	0.86 [0.83,0.88] (n=5,189)

F Additional Ablations

Table 6 reports the full dedicated n=500 ablation sweep over correction-attempt depth and example ordering. Values should be compared within this table rather than directly against Table 1.

The correction-depth sweep shows that the best K is task-dependent. K=1 closely tracks zero-shot because it extracts no rules. Moderate attempt budgets perform best on three tasks: K=5 is strongest for SMM4H-MRC and SMM4H-APO, and also gives the best Stigma-CN result, while N2C2-MC continues to improve to K=10. The pattern is non-monotonic rather than uniformly "more attempts is better," suggesting that additional corrections help only when they yield useful rules.

The ordering sweep shows that the sequence of training examples can shift the classifier's operating point. Pos-first ordering helps N2C2-MC and SMM4H-MRC, whereas Neg-first ordering is stronger on SMM4H-APO. On Stigma-CN, Pos-first sharply reduces F_1, while Neg-first remains closer to the default sweep. Relative to Neg-first, Pos-first raises precision but lowers recall on all four tasks: for example, on N2C2-MC precision rises from 0.54 to 0.62 while recall falls from 0.96 to 0.86, and on SMM4H-MRC precision rises from 0.43 to 0.50 while recall falls from 0.72 to 0.62. On Stigma-CN and SMM4H-APO, the same conservative shift suppresses true positives too strongly: Stigma-CN recall falls from 0.99 under Neg-first to 0.24 under Pos-first, and SMM4H-APO recall falls from 0.94 to 0.67.

G RoBERTa Fine-tuning Details

For the supervised reference, we fine-tune a separate `FacebookAI/roberta-base` sequence-classification model for each task and checkpoint. Each model is trained on the first n rows of the task's training CSV, matching the sequential prefixes used by MELD, and is evaluated on the full fixed test set. The same sequential-prefix protocol is used for all tasks; no development-set selection or checkpoint selection is introduced beyond the requested training size.

Tokenization uses the Hugging Face `AutoTokenizer` with trun-

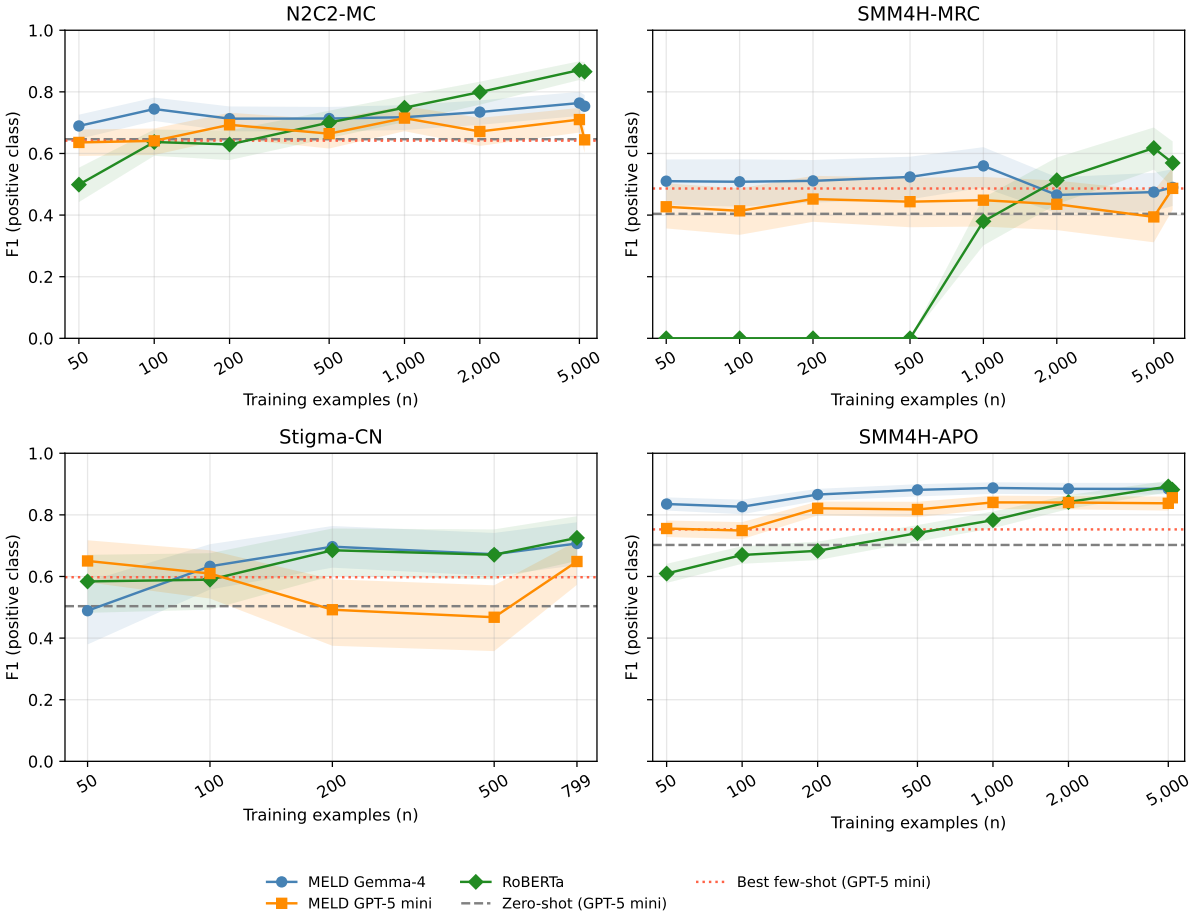


Figure 4: Secondary GPT-5-mini learning curves. The orange curve denotes MELD with GPT-5-mini; the blue curve denotes the primary Gemma-4 MELD run; the green diamond curve denotes RoBERTa fine-tuning. Horizontal reference lines denote GPT-5-mini zero-shot (gray dashed) and best few-shot (red dotted). Shaded regions show 95% bootstrap confidence intervals.

cation, `max_length=512`, and `padding=max_length`. If the tokenizer has no padding token, the script uses the EOS token when available; otherwise it adds `[PAD]` and resizes the model embeddings.

Training uses the Hugging Face `Trainer` API with 10 epochs for $n \leq 200$ and 5 epochs otherwise, train batch size 8, evaluation batch size 64, learning rate 2×10^{-5}, weight decay 0.01, warmup ratio 0.1, the same random seed as the MELD pipeline, and `fp16` when CUDA is available. No development-set evaluation or checkpoint selection is performed during training; temporary trainer directories are removed after evaluation.

Metrics are computed from positive-class precision, recall, and F_1 using the same metric code as MELD, with 1,000 bootstrap resamples for confidence intervals. The main table reports the full-data checkpoint for each task. Figure 2 plots the RoBERTa checkpoints that match available MELD memory snapshots, so the plotted learning curves use comparable training prefixes.

H Licensing and Data Use

This work uses a mixture of openly available model weights, hosted model services, and restricted biomedical datasets. We used each resource under the access conditions made available by its provider, and we do not redistribute model weights, raw clinical notes, raw social media posts, or trained memories derived from restricted text.

Models. The supervised RoBERTa reference uses `FacebookAI/roberta-base`, which is distributed on Hugging Face under an MIT license.[1] The primary Gemma-family backbone was accessed through an institutional hosted server. Use of Gemma-family weights and derivatives is sub-

[1] https://huggingface.co/FacebookAI/roberta-base

Table 4: Class distribution at each learning-curve checkpoint. Train columns report the positive and negative examples in the first n training instances. Test columns report the fixed evaluation-set distribution for the task.

Task	n	Train +	Train -	Train + (%)	Test +	Test -
N2C2-MC	50	11	39	22.00	265	1,047
	100	24	76	24.00		
	200	41	159	20.50		
	500	95	405	19.00		
	1,000	188	812	18.80		
	2,000	392	1,608	19.60		
	5,000	1,012	3,988	20.24		
	5,247	1,062	4,185	20.24		
SMM4H-MRC	50	5	45	10.00	132	1,362
	100	9	91	9.00		
	200	19	181	9.50		
	500	38	462	7.60		
	1,000	79	921	7.90		
	2,000	168	1,832	8.40		
	5,000	440	4,560	8.80		
	5,976	524	5,452	8.77		
Stigma-CN	50	21	29	42.00	88	112
	100	40	60	40.00		
	200	81	119	40.50		
	500	218	282	43.60		
	799	351	448	43.93		
SMM4H-APO	50	21	29	42.00	585	713
	100	45	55	45.00		
	200	92	108	46.00		
	500	234	266	46.80		
	1,000	442	558	44.20		
	2,000	898	1,102	44.90		
	5,000	2,252	2,748	45.04		
	5,189	2,337	2,852	45.04		

ject to the applicable Gemma license or terms attached to the hosted model, including the Gemma Terms of Use when those terms apply.[2] The GPT-5-mini experiments were conducted through the OpenAI API; model access, inputs, and outputs are governed by OpenAI's API documentation and service terms rather than by an open-weight model license.[3] No GPT-5-mini weights were downloaded, modified, or redistributed.

Data. N2C2-MC is derived from the 2022 n2c2 Track 1 Contextualized Medication Event Dataset, which consists of de-identified clinical notes and was distributed to approved users under an n2c2 data use agreement for research use. We report only aggregate metrics and do not reproduce note text. Stigma-CN is distributed through PhysioNet as a restricted-access resource under the PhysioNet Credentialed Health Data License 1.5.0 and requires credentialed access, required training, and a signed data use agreement.[4] We used it under those access conditions and do not redistribute the underlying sentences. SMM4H-MRC and SMM4H-APO are tweet-derived shared-task datasets obtained through the SMM4H data-access process.[5] Their use and redistribution are constrained by the shared-task conditions and applicable Twitter/X developer policies;[6] in particular, raw social media text should not be redistributed outside the permissions granted by those terms. Accordingly, this paper reports aggregate results and prompt templates but does not publish raw tweets.

Derived artifacts. MELD produces persistent natural-language memory files from training disagreements. Because those memories are derived from restricted biomedical or social media data, they may inherit restrictions from the source datasets even when individual rules are phrased generally. Public releases should therefore exclude memories trained on restricted data unless they have undergone privacy review and their redistribution is permitted by the relevant data license or data use agreement. For this reason, the planned public code release is limited to implementation code and synthetic demonstration material.

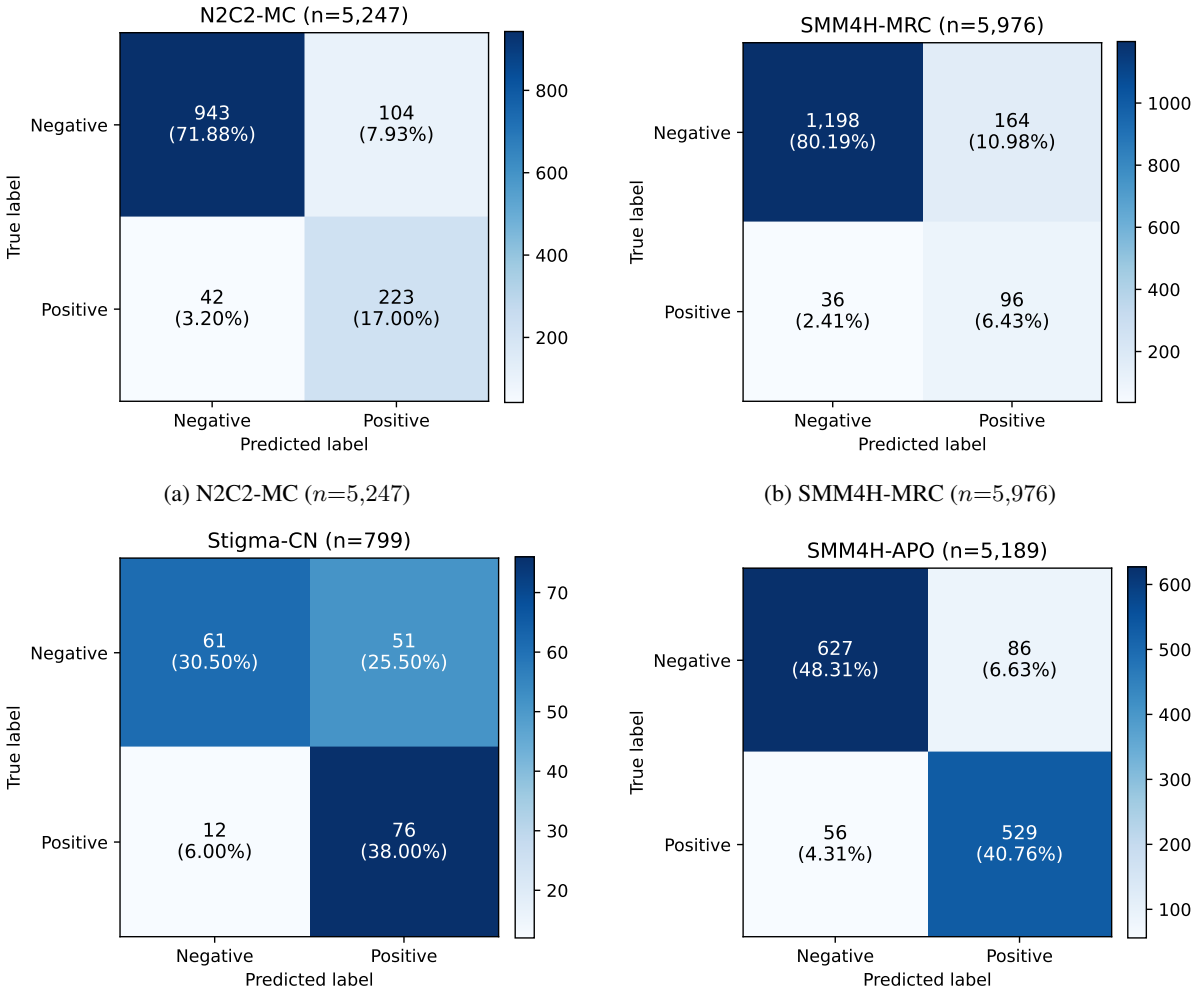
[2] https://ai.google.dev/gemma/terms

[3] https://openai.com/policies/service-terms/

[4] https://www.physionet.org/content/stigmatizing-language/1.0.0/

[5] https://healthlanguageprocessing.org/smm4h-shared-task-2021/

[6] https://docs.x.com/developer-terms/policy



(a) N2C2-MC (n=5,247)

(b) SMM4H-MRC (n=5,976)

(c) Stigma-CN (n=799)

(d) SMM4H-APO (n=5,189)

Figure 5: Confusion matrices (MELD, full training-size checkpoints).

I Computational Budget and Infrastructure

MELD is a gradient-free API-based procedure. The primary Gemma-family experiments and secondary GPT-5-mini experiments were run through hosted OpenAI-compatible APIs, so no local model weights were trained or updated for MELD. Consequently, MELD's training budget is measured in API calls rather than GPU hours. For the latest full-data Gemma-family training runs, the realized budgets were 6,856 API calls for N2C2-MC, 8,080 for SMM4H-MRC, 1,306 for Stigma-CN, and 6,476 for SMM4H-APO, for a total of 22,718 API calls across the four full-data runs. Held-out MELD inference uses one API call per test example.

The supervised reference model, `FacebookAI/roberta-base`, has approximately 125M parameters. RoBERTa fine-tuning used the Hugging Face `Trainer` API with CUDA acceleration and mixed precision (`fp16`) when CUDA was available, as described in Appendix G. The exact parameter counts for the institutional Gemma-family model served as `gemma-4` and for GPT-5-mini are not disclosed by the hosted-model providers; we therefore report the model identifiers and access mode rather than unavailable parameter counts.

J Task Configuration Details

Each task is configured with a classification system prompt, a user prompt template, positive and negative indicator phrases for label matching, two gold-label correction messages, and a rule-extraction prompt. At inference and during training, the user prompt inserts the task input in place of {`text`}. During rule extraction, {`verdict`} is replaced by the verbalized gold label.

Tables 7–9 organize the exact prompt content by

Table 5: Memory statistics at selected checkpoints. Rule rate: percentage of training examples that yielded an extracted rule. Non-conv.: percentage of training examples that never reached the correct label within $K{=}10$ classification attempts. Unmatched: percentage of evaluation outputs that matched no indicator phrase.

Task	n	Rules	Words	Avg. words/rule	Rule rate (%)	Non-conv. (%)	Unmatched (%)
N2C2-MC	50	4	119	29.75	8.00	0.00	0.00
	200	23	685	29.78	11.50	0.00	0.00
	1,000	135	3,951	29.27	13.50	0.40	0.00
	5,247	701	20,339	29.01	13.36	0.44	0.00
SMM4H-MRC	50	7	167	23.86	14.00	8.00	0.00
	200	18	457	25.39	9.00	4.00	0.00
	1,000	122	3,245	26.60	12.20	1.20	0.00
	5,976	971	25,669	26.44	16.25	0.30	0.00
Stigma-CN	50	6	149	24.83	12.00	6.00	0.00
	200	47	1,164	24.77	23.50	1.50	0.00
	500	136	3,338	24.54	27.20	0.60	0.00
	799	240	5,860	24.42	30.04	0.38	0.00
SMM4H-APO	50	5	115	23.00	10.00	2.00	0.00
	200	19	505	26.58	9.50	2.50	0.00
	1,000	119	3,103	26.08	11.90	1.00	0.00
	5,189	549	14,168	25.81	10.58	0.40	0.00

Table 6: Additional ablation results (F_1) from a dedicated $n{=}500$ sweep with 95% bootstrap confidence intervals in brackets. Pos-first indicates that the training data was sorted with positive labels first, while Neg-first denotes the opposite ordering.

Config	N2C2-MC	SMM4H-MRC	Stigma-CN	SMM4H-APO
Max classification attempts K				
$K{=}1$	0.64 [0.60,0.68]	0.41 [0.33,0.47]	0.67 [0.57,0.75]	0.75 [0.72,0.77]
$K{=}3$	0.71 [0.66,0.75]	0.48 [0.40,0.57]	0.65 [0.58,0.72]	0.85 [0.83,0.87]
$K{=}5$	0.72 [0.67,0.75]	**0.54** [0.47,0.61]	**0.68** [0.61,0.75]	**0.87** [0.85,0.89]
$K{=}10$	**0.72** [0.68,0.76]	0.50 [0.42,0.56]	0.65 [0.58,0.72]	0.84 [0.82,0.86]
Example ordering				
Pos-first	**0.72** [0.68,0.76]	**0.56** [0.49,0.62]	0.34 [0.23,0.44]	0.76 [0.73,0.79]
Neg-first	0.69 [0.65,0.72]	0.54 [0.47,0.60]	**0.65** [0.58,0.71]	**0.87** [0.85,0.89]

prompt role.

Table 7: System prompts used for classification and rule extraction.

Task	Classification system prompt	Rule-extraction system prompt
N2C2-MC	You are a clinical NLP expert. Given a clinical note, determine whether a medication change is discussed. Answer with exactly: "the presence of a medication change is discussed" or "the presence of a medication change is not discussed".	You are a clinical NLP expert. Given a clinical note and its correct classification, extract ONE concise generalizable rule that helps identify whether a medication change is discussed. Frame it as: "When a clinical note [contains/lacks] [specific pattern], a medication change [is/is not] discussed." Do not include patient-specific details.
SMM4H-MRC	You are a social media NLP expert. Given a tweet, determine whether the user is self-reporting a change in medication regimen. Answer with exactly: "the tweet is self reporting a change in medication regimen" or "the tweet is not self reporting a change in medication regimen".	You are a social media NLP expert. Given a tweet and its correct classification, extract ONE concise generalizable rule that helps identify whether the user is self-reporting a change in medication regimen. Frame it as: "When a tweet [mentions/contains] [specific pattern], the user [is/is not] self-reporting a medication change." Do not include user-specific details.
Stigma-CN	You are a clinical NLP expert. Given a clinical note sentence, determine whether it involves language about the patient that could result in stigmatization or negative labeling, which could lead to further status loss or discrimination in the patient-provider relationship. Answer with exactly: "the note involves stigmatization language" or "the note does not involve stigmatization language".	You are a clinical NLP expert. Given a clinical note sentence and its correct classification, extract ONE concise generalizable rule that helps identify whether a note involves stigmatization language. Frame it as: "When a clinical note [contains/uses] [specific pattern], it [does/does not] involve stigmatization language." Do not include patient-specific details.
SMM4H-APO	You are a social media NLP expert. Given a tweet, determine whether the user is self-reporting an adverse pregnancy outcome. Answer with exactly: "the tweet is self reporting an adverse pregnancy outcome" or "the tweet is not self reporting an adverse pregnancy outcome".	You are a social media NLP expert. Given a tweet and its correct classification, extract ONE concise generalizable rule that helps identify whether the user is self-reporting an adverse pregnancy outcome. Frame it as: "When a tweet [mentions/contains] [specific pattern], the user [is/is not] self-reporting an adverse pregnancy outcome." Do not include user-specific details.

Table 8: User prompt templates and label-matching phrases.

Task	User template	Positive indicators	Negative indicators
N2C2-MC	Clinical note: {text}.	"the presence of a medication change is discussed"; "presence of a medication change is discussed"; "medication change is discussed".	"the presence of a medication change is not discussed"; "presence of a medication change is not discussed"; "medication change is not discussed".
SMM4H-MRC	Tweet: {text}.	"the tweet is self reporting a change in medication regimen"; "is self reporting a change in medication regimen"; "self reporting a change in medication regimen".	"the tweet is not self reporting a change in medication regimen"; "is not self reporting a change in medication regimen"; "not self reporting a change in medication regimen".
Stigma-CN	Clinical note: {text}.	"the note involves stigmatization language"; "note involves stigmatization"; "involves stigmatization".	"the note does not involve stigmatization language"; "note does not involve stigmatization"; "does not involve stigmatization".
SMM4H-APO	Tweet: {text}.	"the tweet is self reporting an adverse pregnancy outcome"; "is self reporting an adverse pregnancy outcome"; "self reporting an adverse pregnancy outcome".	"the tweet is not self reporting an adverse pregnancy outcome"; "is not self reporting an adverse pregnancy outcome"; "not self reporting an adverse pregnancy outcome".

Table 9: Gold-label correction messages and rule-extraction user templates.

Task	Positive correction	Negative correction	Rule-extraction user template
N2C2-MC	That is not correct. The clinical note does discuss a medication change. Please answer with exactly: "the presence of a medication change is discussed".	That is not correct. The clinical note does not discuss a medication change. Please answer with exactly: "the presence of a medication change is not discussed".	Clinical note: {text}. Correct classification: {verdict}. Write one generalizable rule in one sentence.
SMM4H-MRC	That is not correct. The tweet does self-report a change in medication regimen. Please answer with exactly: "the tweet is self reporting a change in medication regimen".	That is not correct. The tweet does not self-report a change in medication regimen. Please answer with exactly: "the tweet is not self reporting a change in medication regimen".	Tweet: {text}. Correct classification: {verdict}. Write one generalizable rule in one sentence.
Stigma-CN	That is not correct. The note does involve stigmatization language. Please answer with exactly: "the note involves stigmatization language".	That is not correct. The note does not involve stigmatization language. Please answer with exactly: "the note does not involve stigmatization language".	Clinical note: {text}. Correct classification: {verdict}. Write one generalizable rule in one sentence.
SMM4H-APO	That is not correct. The tweet does self-report an adverse pregnancy outcome. Please answer with exactly: "the tweet is self reporting an adverse pregnancy outcome".	That is not correct. The tweet does not self-report an adverse pregnancy outcome. Please answer with exactly: "the tweet is not self reporting an adverse pregnancy outcome".	Tweet: {text}. Correct classification: {verdict}. Write one generalizable rule in one sentence.