ACSS-PSL at #SMM4H-HeaRD 2026: An LLM-Driven Autoresearch Loop for Opioid-Impact NER

Olivier Caron and Bruno Chaves Ferreira and Christophe Benavent
DRM, Université Paris-Dauphine, PSL University, CNRS, 75016 Paris, France
{olivier.caron, bruno.chavesferreira, christophe.benavent}@dauphine.psl.eu

Abstract

We apply an LLM-driven autoresearch protocol to Task 7 of #SMM4H-HeaRD 2026, which requires extracting *ClinicalImpacts* and *SocialImpacts* spans from Reddit posts about non-medical opioid use. A coding agent iteratively proposes a hypothesis, modifies the training configuration, and evaluates against the held-out validation set. Across 79 runs, only 9 improved strict F1, indicating a narrow viable search space on this small dataset (842 training examples). The submitted ensemble combines DeBERTa-large, MC Dropout blending, and a constrained multi-LLM consensus layer, reaching 0.46 strict and 0.52 relaxed F1 on test, though single-seed evaluation limits the reliability of run-level comparisons. The run log provides a reproducible case study of autonomous experimentation, including failure modes and guardrails for small-data NER.

1 Introduction

NER shared-task development is usually manual and rarely logged in detail. This paper evaluates an alternative: an LLM agent that iterates through propose-train-evaluate-update cycles, following recent work on autonomous scientific discovery (Karpathy, 2026; Lu et al., 2024; Yamada et al., 2025; Schmidgall et al., 2025), agentic AutoML (Trirat et al., 2025; Chi et al., 2024), and benchmarks for LLM-driven ML experimentation (Huang et al., 2024; Chan et al., 2025).

We apply this approach to Task 7 of #SMM4H-HeaRD 2026 (Lopez-Garcia et al., 2026), which focuses on extracting self-reported *ClinicalImpacts* (e.g., withdrawal, overdose) and *SocialImpacts* (e.g., job loss, legal issues) from Reddit posts about non-medical opioid use (Dey et al., 2026). The official split contains 842 training and 258 validation examples. The training set includes 256 *ClinicalImpacts* and 87 *SocialImpacts* spans; the validation set includes 92 and 27, respectively. Both sets reflect a roughly 3:1 class imbalance.

The contributions are: (1) a reproducible autoresearch workflow with a 79-run audit trail, (2) evidence that most standard techniques degraded performance under this budget and dataset, and (3) practical guardrails and observed failure modes for autonomous NER experimentation.

2 Autoresearch Protocol

2.1 Loop Architecture

The autoresearch loop uses the Claude Code CLI agent[1] (Claude Opus 4.6) with persistent repository access. Each run follows six steps: load state, generate one hypothesis, edit a configuration block, train (∼15–20 min), evaluate (strict/relaxed F1), and update logs/state.

Each experiment has an ID and a structured log entry (timestamp, hypothesis, hyperparameters, class-wise metrics, accept/reject). We only intervened for GPU restarts and occasional re-prompts after long runs.

Following the single-file principle of Karpathy's autoresearch (Karpathy, 2026), the agent was instructed to modify only `train.py`, keeping the scope manageable and the context window focused; data loading, evaluation, and scoring scripts remained untouched throughout. Given that the task organizers' baseline used `deberta-large` (Dey et al., 2026), the agent started directly with this backbone. No other instructions constrained the search space: the agent freely explored hyperparameters, training loss, probability thresholds, and post-processing. The only hard constraint was the available GPU (RTX 3070, 8 GB VRAM), which ruled out larger models. The agent independently proposed the LLM consensus ensemble; we kept it after review. External API access was initially under-constrained (see below).

[1]Claude Code: https://docs.anthropic.com/en/docs/claude-code

Proceedings of the 11th Social Media Mining for Health Research and Applications (SMM4H-HeaRD 2026) Workshop and Shared Tasks, pages 240–245

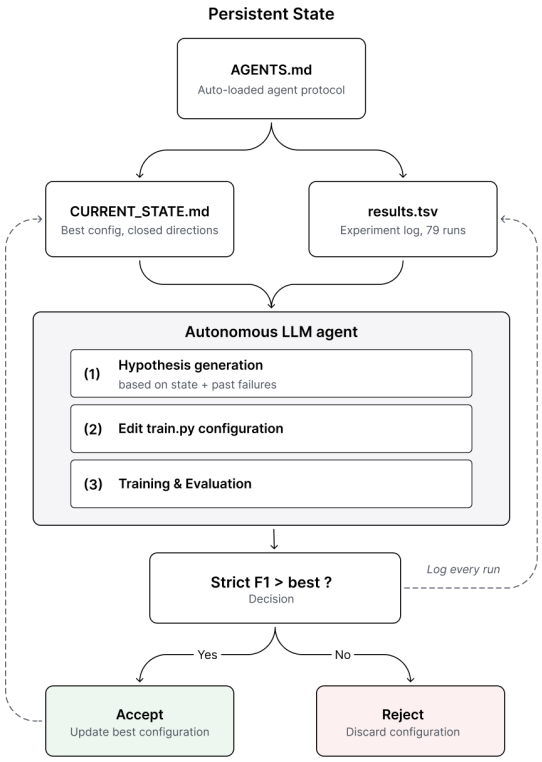

Figure 1: Autoresearch loop. `CURRENT_STATE.md` stores the best configuration and closed directions. `results.tsv` logs every run. Each iteration reads state, edits `train.py`, trains and evaluates, and updates state only when strict F1 improves.

2.2 Credential-Use Incident

The agent initiated paid LLM API calls without explicit prompting, because API credentials were available in the execution environment. The initial protocol used prompt instructions, not code checks, to prevent external calls. After the incident, LLM access was restricted to an explicit allowlist of five models with per-session usage caps. This is only a partial fix. Future loops should block unauthorized calls in the sandbox, not only through prompt rules.

2.3 State Management and Heuristics

Two files store the agent's memory across context-window resets. The experiment log `results.tsv` stores one row per run with timestamp, hypothesis, hyperparameters, class-wise metrics, and accept/reject status. The state document `CURRENT_STATE.md` follows Karpathy's `state.md` pattern (Karpathy, 2026): it stores the best configuration, closed directions with supporting evidence, and prioritized next steps.

The agent applies four heuristics: monotonic acceptance (strict F1 must improve), direction closing after repeated failures, minimal attributable edits between accepted runs, and diagnostic-driven hypothesis selection. However, accept/reject decisions were based on single-seed validation F1, and seed variance spans 0.09 F1 points (Section 4). Individual decisions therefore conflate random effects with hypothesis quality. N-fold cross-validation would be a more robust procedure for this kind of loop, especially with such a small and noisy dataset. The run history is thus a fixed-budget model-selection record, not reliable evidence that each accepted change improved generalization.

2.4 Reproducibility

Strict and relaxed F1 are computed with the official task scorer (Dey et al., 2026) on character-offset spans. Post-processing applies in fixed order: (i) per-class probability thresholds, (ii) boundary trimming, (iii) first-person pronoun filter. Every run is identified by an experiment ID; the seed and `results.tsv` row specify the exact conditions.

3 System Description

3.1 DeBERTa-Large Token Classifier

We frame the task as BIO token-level sequence labeling with five labels (`B/I-ClinicalImpacts`, `B/I-SocialImpacts`, `O`). Prior SMM4H span-detection work has applied pre-trained transformers successfully (Guo et al., 2021). The core system is a `microsoft/deberta-large` token classifier (He et al., 2021) (304M parameters). Best configuration after 79 runs: learning rate 1×10^{-5}, cosine schedule, 10% warmup, effective batch size 8, 10 epochs, seed 42, max length 192, unweighted cross-entropy. The agent also explored larger variants: `deberta-xlarge` with LoRA underperformed under the 8 GB GPU limit, and `deberta-v2-xlarge` with full fine-tuning converged poorly.

Three post-processing steps improve predictions: (1) per-class probability thresholds (0.5 for *ClinicalImpacts*, 0.35 for *SocialImpacts*), (2) boundary trimming to remove leading articles and trailing punctuation, and (3) a first-person pronoun filter to suppress predictions in posts lacking self-referential language.

3.2 Multi-Source Ensemble

DeBERTa tuning plateaued at 0.492 strict F1 for the base model and 0.498 with MC Dropout blend-

Direction	Runs	Best F1	Δ
Seed variants	3	0.492	0.000
Other	12	0.486	−0.006
Epochs (4–15)	6	0.485	−0.007
Post-processing	5	0.485	−0.007
Regularization	8	0.481	−0.011
Data augment.	3	0.467	−0.025
Synth. data	6	0.458	−0.034
Loss functions	5	0.453	−0.039
Learning rate	4	0.452	−0.040
Alt. backbones	7	0.407	−0.085
Total	59		

Table 1: Best validation strict F1 by research direction across 59 DeBERTa runs. Baseline (A17) reaches 0.492. No training-time variation beat the baseline; validation gains come entirely from inference-time blends (MC Dropout, LLM consensus).

ing (Gal and Ghahramani, 2016). The agent then proposed a few-shot LLM consensus layer. LLMs achieve high relaxed but low strict F1, while DeBERTa shows the reverse pattern: their strengths are complementary. Structured prompting has independently shown gains in biomedical NER (Ge et al., 2026), though the agent did not explicitly reference prior work. The ensemble operates in three stages. First, standard DeBERTa and MC Dropout predictions (20 forward passes, α=0.85 MC + 0.15 standard logits) are converted to character-offset spans. Second, five LLMs produce JSON span predictions using the same few-shot prompt (Appendix A). Third, an LLM span is retained only when it overlaps a DeBERTa span and receives at least two LLM votes; when all five LLMs agree on a 1-token wider boundary, the DeBERTa boundary is overridden. On validation, strict F1 improves from 0.492 (DeBERTa-only) to 0.498 (MC blend) to 0.565 (full ensemble).

4 Results and Analysis

4.1 Exploration Summary

Across 79 runs, only 9 strictly improved on the running best strict F1 (∼11.4%). Table 1 categorizes the 59 DeBERTa-focused runs by research direction.[2] Figure 2 shows the trajectory: the curve plateaus early, and dashed ellipses highlight three failure clusters (synthetic data, regularization, LLM zero/few-shot baselines).

[2]The remaining 20 runs (LLM-only baselines, ensembles, diagnostic/model-save runs) are not categorized.

System	Strict	Relaxed
Validation (258 examples)		
DeBERTa (A17)	0.492	0.552
+ MC blend	0.498	0.580
+ LLM ensemble	0.565	0.700
Test		
Submitted ensemble	0.46	0.52
All teams mean	0.46	0.55
All teams median	0.48	0.58

Table 2: System performance on validation and test sets.

4.2 Key Findings

Most techniques degraded performance (Table 1). Explored ranges included learning rates from 5×10^{-6} to 3×10^{-5}, 4–15 epochs, weight decay 0.0–0.1, label smoothing up to 0.1, and four loss functions (cross-entropy, weighted CE, focal, CRF). Seven alternative backbones all stayed below 0.41 versus 0.492 for DeBERTa-large. Seed sensitivity is substantial. Strict F1 varies by 0.09 across seeds: seed 42 reaches 0.492, seed 1 drops to 0.441, and seed 7 to 0.402. Multi-seed ensembles (0.455) score below the best single seed. The dominant error types are boundary mismatches (DeBERTa predicting spans that are too short or too long), narrator confusion (extracting third-person impacts), and label ambiguity between Clinical and Social on overlap cases such as job loss caused by withdrawal.

4.3 Task Results

Table 2 presents the results. On validation, the MC Dropout blend adds 0.006 strict F1 over base DeBERTa, and the LLM consensus layer adds a further 0.067. On test, the submitted ensemble achieves 0.46 strict and 0.52 relaxed F1, below the median across participating teams (0.48 strict, 0.58 relaxed).[3]

The validation–test gap (0.565 → 0.46) suggests overfitting after repeated evaluation on a 258-example validation set. The LLM consensus layer is particularly susceptible: its voting thresholds and merge rules were selected based on the same validation examples used throughout the loop. Because we did not submit a standalone DeBERTa system, we cannot tell whether the LLM ensemble's validation gain transferred to the blind test set. The test score therefore reflects the full submitted protocol, not evidence that each component improved out-of-sample.

[3]From official Task 7 results communicated to participants.

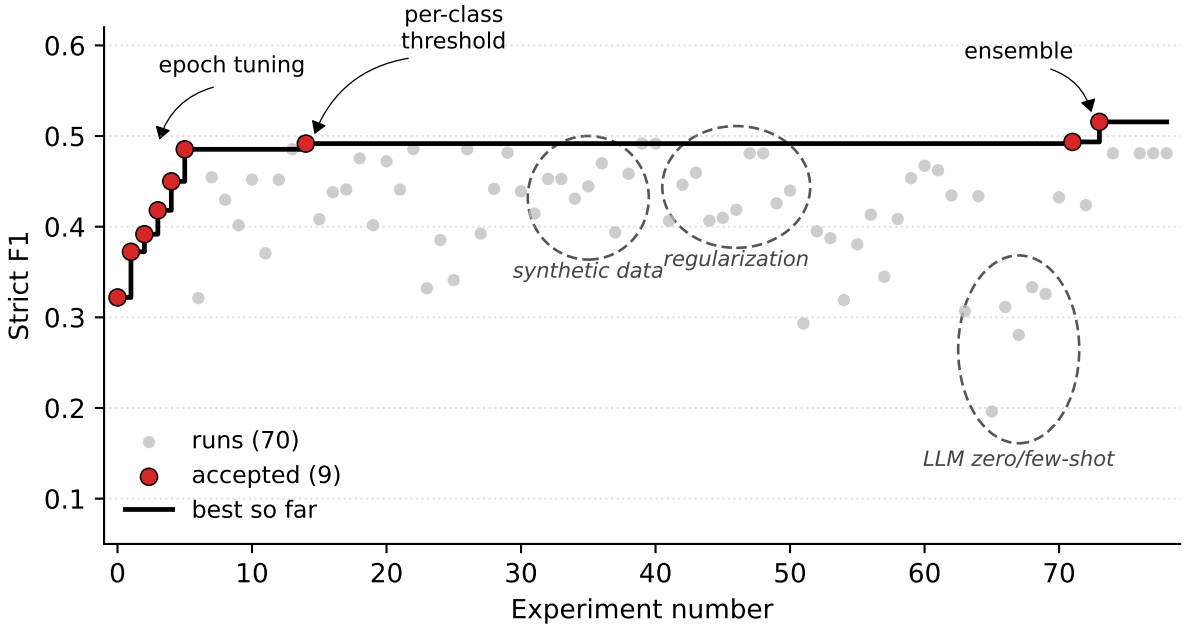

Figure 2: Exploration trajectory across 79 runs. Gray dots are individual experiments; red dots mark the 9 accepted improvements. The black staircase tracks the running best strict F1. Dashed ellipses highlight failure clusters.

5 Conclusion

The autoresearch loop systematically explored 79 configurations with a complete audit trail, requiring approximately 25 hours of GPU time; human intervention was limited to GPU restarts and occasional re-prompts. The submitted system did not beat the task median. The primary finding is methodological: most standard NLP techniques the agent tried degraded performance, and the viable configuration space proved narrow. This approach offers systematic coverage and full logging, but needs safeguards: monotonic acceptance, direction closing, structured state management, and upfront rules for external API usage (allowed models, budget caps). It also needs stronger validation (e.g., cross-validation) than we used here. Granting the agent access to the scientific literature (e.g., arXiv, Semantic Scholar) could broaden its hypothesis space beyond what its training data encodes.

Limitations

As discussed in Sections 2 and 4, single-seed evaluation and repeated optimization on a 258-example validation set confound random effects with hypothesis quality; we did not run cross-validation, a matched-compute manual baseline, or a DeBERTa-only test submission. Experiments were constrained by a single RTX 3070 (8 GB), and hypothesis generation may reflect LLM prior biases. The LLM ensemble introduces API cost and reproducibility constraints: parameter counts for proprietary LLMs are not publicly disclosed, and we report model identifiers only. The credential-use incident (Section 2) shows that prompt-level instructions are insufficient for autonomous loops; future work should enforce resource limits in the sandbox.

References

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. 2025. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*.

Yizhou Chi, Yizhang Lin, Sirui Hong, Duyi Pan, Yaying Fei, Guanghao Mei, Bangbang Liu, Tianqi Pang, Jacky Kwok, Ceyao Zhang, Bang Liu, and Chenglin Wu. 2024. SELA: Tree-search enhanced LLM agents for automated machine learning. *arXiv preprint arXiv:2410.17238*.

Sumon Kanti Dey, Jeanne M. Powell, Azra Ismail, Jeanmarie Perrone, and Abeed Sarker. 2026. Inference gap in domain expertise and machine intelligence in named entity recognition: Creation of and insights from a substance use-related dataset. In *Proceedings of the Pacific Symposium on Biocomputing*. World Scientific.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Yao Ge, Yuting Guo, Sudeshna Das, and Abeed Sarker. 2026. Improving few-shot named entity recognition for large language models using structured dynamic prompting with retrieval augmented generation. *npj Artificial Intelligence*, 2(1):39.

Yuting Guo, Yao Ge, Mohammed Ali Al-Garadi, and Abeed Sarker. 2021. Pre-trained transformer-based classification and span detection models for social media health applications. In *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*, pages 52–57, Mexico City, Mexico. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2024. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 20271–20309. PMLR.

Andrej Karpathy. 2026. Autoresearch: LLM-driven autonomous research. `https://github.com/karpathy/autoresearch`. Open-source framework for autonomous ML experimentation.

Guillermo Lopez-Garcia, Jose Miguel Acitores Cortina, Jacob Berkowitz, Joey Chan, Ganesh Chandrasekar, Sumon Kanti Dey, Ivan Flores Amaro, Fernando Gallego, Lauren Gryboski, Ari Z Klein, Martin Krallinger, Salvador Lima-López, Tomohiro Nishiyama, Lisa Raithel, Ahmad Rezaie Mianroodi, Roland Roller, Judith Rosell, Frank Rudzicz, Abeed Sarker, and 8 others. 2026. Overview of the 11th social media mining for health (#smm4h) and health real-world data (HeaRD) shared tasks at ACL 2026. In *Proceedings of the 11th Social Media Mining for Health (#SMM4H) and Health Real-World Data (HeaRD) Workshop and Shared Tasks*. Association for Computational Linguistics.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using LLM agents as research assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5977–6043, Suzhou, China. Association for Computational Linguistics.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. 2025. AutoML-agent: A multi-agent LLM framework for full-pipeline AutoML. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*.

A LLM Consensus Layer Details

The submitted ensemble queries five LLMs with the same system prompt and three hand-picked few-shot examples covering three cases: a post with both *Clinical* and *Social* spans, a post with multiple *Clinical* spans, and a post with no spans. We set temperature to 0 for deterministic models; `gpt-5.4-nano` uses default sampling.

The five models are:

- `gpt-4.1-mini`
- `gpt-4.1-nano`
- `gpt-5.4-nano` (snapshot 2026-03-17)
- `gemini-2.5-flash` (Vertex AI)
- `gemini-3-flash-preview` (Vertex AI)

We merge predictions by token-overlap voting: we keep an LLM span when it overlaps a DeBERTa span and at least two of the five LLMs agree. When all five LLMs agree on a 1-token wider boundary, the DeBERTa boundary is overridden.

A.1 Few-Shot Examples

Table 3 lists the three demonstrations.

A.2 System Prompt

The shared system prompt is reproduced verbatim below.

```
You extract SELF-REPORTED drug-related
impact entities from first-person Reddit
posts about opioid/substance use.

These are personal narratives where the
author describes impacts they PERSONALLY
experienced from their own drug use.

ENTITY TYPES:
- ClinicalImpacts: health/medical effects
  the author experienced - withdrawal,
  overdose, physical symptoms, mental
  health effects, addiction, tolerance,
  cravings, pain, intoxication effects,
  rehab, detox, MAT treatment
- SocialImpacts: life/social consequences
  the author experienced - relationship
  problems, job loss, financial issues,
  legal trouble, homelessness, isolation,
  family conflict, arrests
```

#	Post	Gold annotations
1	If I was super dependent on it, wouldn't I be feeling some semblance of withdrawal every day before I took my dose?	Social: "super dependent on it". Clinical: "withdrawal", "my dose".
2	I went into drug-induced psychosis, which is honestly the scariest thing I have ever experienced – and I am so lucky that I snapped out of the psychotic episode and went back to being my 'self' who I am today.	Clinical: "drug-induced psychosis", "psychotic episode".
3	Subs linger around a loonnggg time!	(no entities)

Table 3: Three few-shot demonstrations. Posts are sent as user messages; gold annotations (JSON array of `{text, type}` objects) as assistant replies.

```
CRITICAL RULES:
1. Each "text" MUST be an EXACT verbatim
   substring of the post - copy-paste
   precision
2. Keep spans MINIMAL: extract only the
   core impact phrase, not surrounding
   context
3. Do NOT include pronouns like "I",
   "my", "me" at the start of spans
4. Do NOT include conjunctions like
   "and", "to", "of" at span boundaries
5. Only extract impacts CAUSED BY
   drug/substance use that the AUTHOR
   personally experienced
6. Do NOT extract impacts described
   about OTHER people (e.g., "my friend
   overdosed")
7. Do NOT extract entities from general
   advice, information sharing, or
   questions about drugs
8. If the post is not a first-person
   account of personal drug impacts,
   return: []

OUTPUT: JSON array of {"text": "exact
quote", "type":
"ClinicalImpacts"|"SocialImpacts"}
```