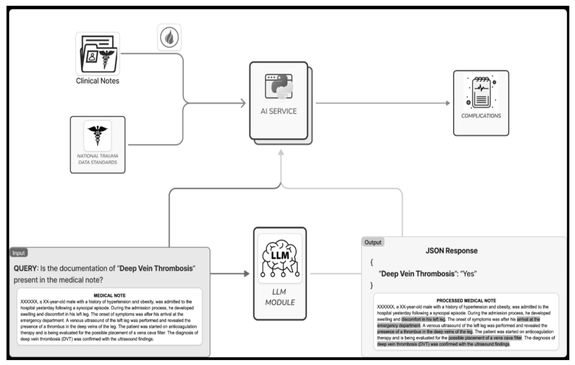
Large Language Models for Trauma Care Quality: NTDS Complication Abstraction

Kaijie Zhang kaz029@ucsd.edu
Viv Somani visomani@ucsd.edu
Mentor: Aaron Boussina aboussina@health.ucsd.edu

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Background & Research Question

- Manual abstraction is slow and inconsistent when applied to long, noisy trauma notes, and the professional staff required for this task are costly.
- Key constraint: secure setting, no external online APIs beyond AWS Bedrock, and no local GPU deployment.
- We compare inference-time strategies under the same retrieval pipeline.
- We ask whether an LLM can detect NTDS complications with auditable note evidence.



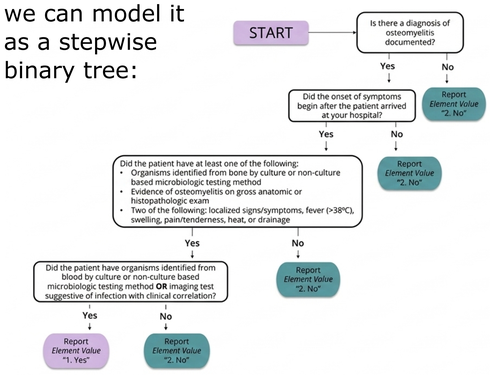
Data Collection

- Built for NTDS/TQIP use: encounter-level patient feature JSON bundles per case. Inputs include trauma notes ("binary") plus medication order text.
- Clinical notes are retrieved from the EHR via FHIR R4 and organized into a unified bundle for processing. The full dataset contains 482 patient features, provided by UCSD Health.
- The NTDS Data Dictionary defines 18 complication -specific prompts used by the AI service. One encounter can contain multiple complications, and notes are abbreviation-heavy and imbalanced.

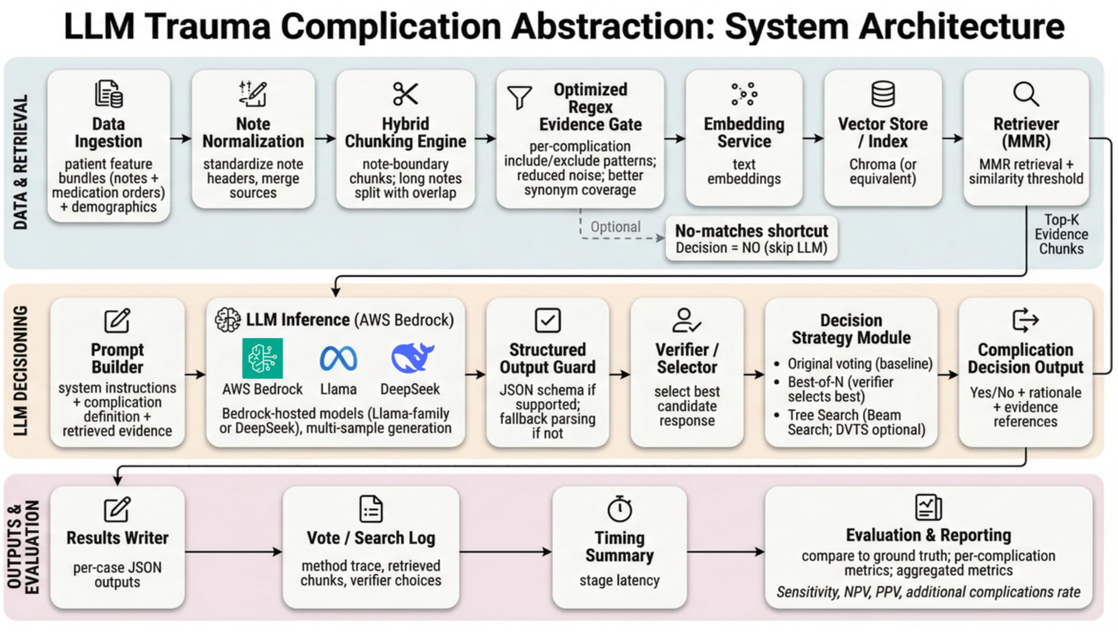
ACS tqip TRAUMA QUALITY IMPROVEMENT PROGRAM UC San Diego Health

Prompt Design

- This is an example full decision path for one complication. Using the NTDS dictionary, we can model it as a stepwise binary tree:

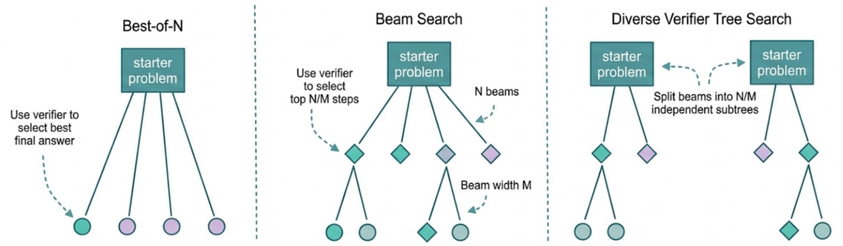


Method Overview & Implementation



Results

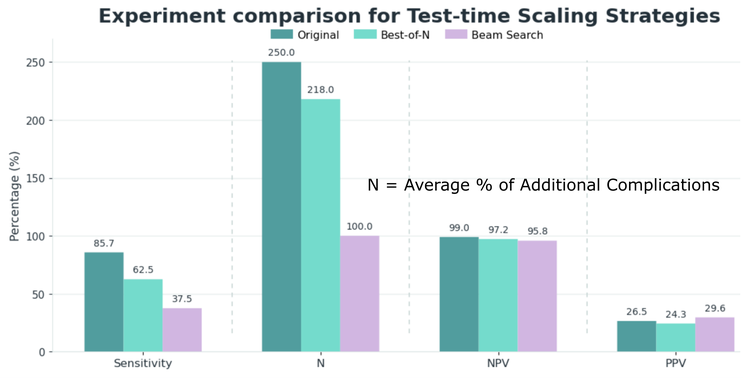
Scaling Methods for Experiment



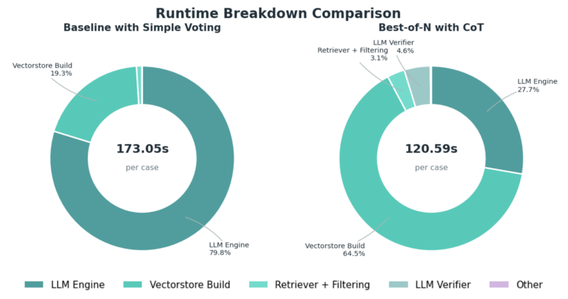
Insight from Results

- Stricter test-time search consistently reduced extra complications, with Beam Search achieving the cleanest output.
- This gain came with a clear recall tradeoff, as sensitivity dropped from Original to Best-of-N and further to Beam Search.
- Best-of-N provided the most balanced middle point, improving cleanliness over Original without becoming as conservative as Beam Search.

Experiment Result



Time Analysis

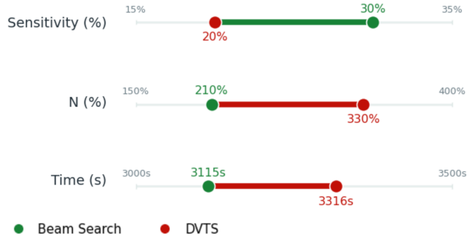


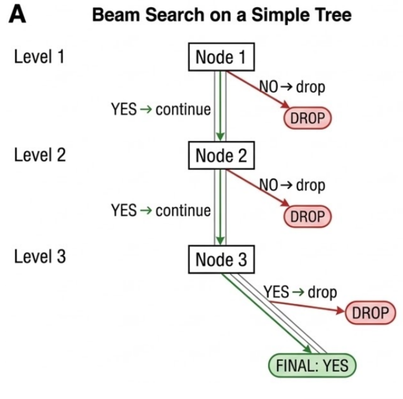
- Best-of-N with CoT is faster per case because runtime shifts away from repeated LLM generation, although vectorstore building becomes the dominant cost.

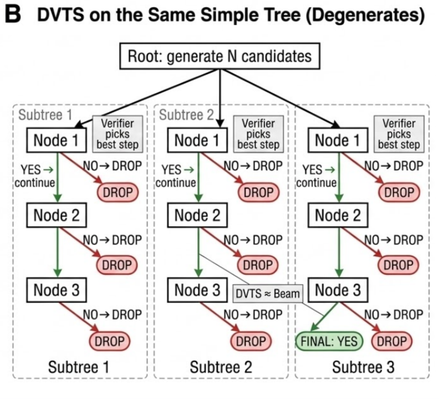
Case Study: DVTS for Ventilator-Associated Pneumonia (VAP)

VAP Case Study between Beam Search and DVTS

On the VAP-enriched subset, the main limitation was recall rather than precision, and beam search preserved the VAP-related path better than DVTS, which spent more search budget on non-VAP complications.







Summary of Findings

- Best-of-N with CoT improved negative-side cleanliness over the baseline and provided the strongest overall trade-off point.
- Beam Search became too strict under the atomic design, which caused some truly positive features to return no complication at all and increased false negatives; DVTS behaved similarly because most complication trees were too simple to benefit from subtree diversity.
- Hybrid chunking produced only a small overall gain, while retrieval quality was more visibly affected by the quality and coverage of complication-specific filters.

Next Step

- Add a Best-of-N fallback when Beam Search or DVTS returns no complication.
- Redesign complication-specific decision trees so DVTS can benefit from real structural diversity.
- Improve and expand complication-specific filters to strengthen retrieval quality.
- Re-design the architecture for better LLM models such as DeepSeek R1.