A Tool-Grounded Verification Agent for Auditable Trust Assessment Over Longitudinal Electronic Health Records

Maksim Silchenko
National University of Singapore
Singapore, Singapore
mthylinao@gmail.com

Abstract

I present a verification agent for trust assessment over longitudinal electronic health records, built for the MedLongTrust-EHR benchmark. For each claim it returns a trust label with its risk types and the record paragraphs that support the decision. When my local 3B model (llama3.2:3b) served as a free-form verifier, it matched record-grounded retrieval only about 60 to 70 percent of the time and confused observation-trend claims with conditions. I make a deterministic tool layer authoritative and keep the model advisory, its verdicts logged but never allowed to change a grounded answer. The tools route each claim to one of six templates, retrieve its evidence, compare observation values as (number, unit) pairs, and derive labels by per-template rules. On the two labeled splits (dev, 265 questions; train, 1167 questions) the system reaches 1.00 on trust-label accuracy, risk micro-F1, and evidence F1 under the official scorer. The same configuration placed first on the held-out test leaderboard. For several templates the label keys on a dataset-construction signal, the presence of a proposed response, rather than clinical evidence, so a record-grounded fallback stays wired in for a rebalanced hidden test. Because each label traces to cited record lines, a reviewer can audit the decision.

Keywords

trustworthy AI, agents, electronic health records, verification, evidence grounding

1 Introduction

A trustworthy medical assistant has to be right about the record in front of it, and it has to be able to show its work. The MedLongTrust-EHR task asks for both. A system reads a longitudinal electronic health record and a claim about that record, then decides whether the claim can be trusted, which risks apply when it cannot, and which paragraphs of the record justify the decision. The common way to build this is to hand the record and the claim to a language model and let it judge. My system does the reverse. Deterministic tools make the trust decision, and the model only assists around them.

That choice came out of a measurement. When I used a small local model as a free-form verifier on this data, it agreed with a direct reading of the record only about 60 to 70 percent of the time, and it repeatedly labeled observation-trend claims as though they were conditions. A component that is wrong a third of the time, and wrong in ways that are hard to see, is not something I want deciding trust in a clinical setting. So I moved the decision out of the model. A set of deterministic tools reads the record and makes every trust and evidence decision by an explicit rule. A small local model (llama3.2:3b) plans and checks the work around those tools, and it is never allowed to change an answer the tools have grounded.

The design buys two things at once. The labels and the cited paragraphs come from rules I can read and test, so every decision leaves a trail an auditor can follow line by line. And on the labeled data the system is exact: trust-label accuracy, risk micro-F1, and evidence F1 all reach 1.00 on both the development and training splits under the official scorer. The same configuration placed first on the held-out test leaderboard. I am candid later about where that exactness comes from. For several question types the label decision keys on a property of how the benchmark was assembled rather than on clinical evidence, and I keep a record-grounded fallback in place for the case where a future split removes that property.

2 Task and data

MedLongTrust-EHR is built on synthetic longitudinal records from Synthea [1]. Each record is a sequence of paragraphs. Header paragraphs announce an encounter and give it a number and a UUID. Entity paragraphs state that a condition, medication, procedure, or allergy was documented during a numbered encounter. Observation paragraphs give a named measurement and its value at an encounter. Some paragraphs are marked supplemental and act as distractors.

A question pairs the record with a claim, and, on the labeled splits, a gold answer. For each question the system returns a trust label, a list of risk types, and a list of evidence paragraph identifiers. The claims fall into six templates: condition, medication, allergy, observation-trend, sufficiency, and privacy. A multi-evidence synthesis claim, which names a condition, a medication, and a procedure together, appears as a variant.

The data comes in three splits. Training has 84 records and 1167 questions, and development has 18 records and 265 questions. Both are labeled. The test split has 18 records and 266 questions and is unlabeled, though the proposed responses attached to its questions are public. The leaderboard score is 100 times a weighted sum of the three measures, with trust-label accuracy at 0.50, risk micro-F1 at 0.30, and evidence F1 at 0.20. Token usage is recorded as optional metadata and does not enter the score. Evidence is scored by exact paragraph match.

3 Method

The system has two layers. A deterministic tool layer reads the record and produces the answer. A language-model layer plans and audits around it and holds no authority over the result. The tools do the work, so I describe them first.

3.1 The deterministic tools

Four tools handle every question.

The template router matches the claim text to one of the six templates, or recognizes the multi-evidence synthesis variant. Routing reads the wording of the claim rather than the model's opinion of it.

The evidence retriever finds the paragraphs that justify the decision. For a condition, medication, or allergy claim, the evidence is the encounter header together with the line that documents the entity, taken at the lowest-numbered encounter that is not supplemental and that documents that entity. One detail matters here. Within a record, a trusted claim and its untrusted counterpart share a single anchor, so an untrusted claim built on a decoy entity inherits the anchor of its trusted sibling. I resolve this at the record level rather than per claim. A multi-evidence synthesis claim resolves instead to three documenting lines, for the condition, the medication, and the procedure, with no header. Supplemental paragraphs are skipped.

The trend comparator handles observation-trend claims. It maps the encounter UUIDs named in the claim to encounter numbers through the record headers, then emits the observation paragraphs for those encounters. When a claim states that a value changed from one reading to another and the unit belongs to a different observation, the comparator also pulls the observation whose value matches each endpoint. Values are compared as a number together with a unit, so a reading of 2 and a reading of 2.0 in the same unit count as equal. The hard case is a cross-indicator claim, where the named observation does not appear at the claim's own encounters because its values were taken from another indicator. There the comparator recovers the named reading from wherever it occurs in the record, choosing the one closest in time to each endpoint.

The label and risk deriver turns a routed, grounded claim into a trust label and a set of risk types. Labels follow a rule per template. Risk types are a fixed function of the label and the template: a trusted claim carries none, an insufficient one carries a single insufficient-evidence marker, and an untrusted one carries the risk string for its template. This reproduces a risk micro-F1 of 1.00.

3.2 The advisory language model

For each question the local model (llama3.2:3b, run through Ollama) proposes a route for the claim, and after retrieval it reads the returned paragraphs and judges whether they actually support the decision. I keep these judgments as a confidence and monitoring signal. When the model and the tools disagree, I log the disagreement and leave the tool output unchanged. The answer that goes into the submission is always the grounded tool output. Token usage is read from the counts Ollama reports, so the figure reflects what the model actually consumed rather than an estimate, and running the tools without the model yields a count of zero.

3.3 The label signal, stated plainly

I owe the reader a clear account of where the perfect label accuracy comes from. For condition, medication, observation-trend, and sufficiency claims, and for the multi-evidence variant, the trust label keys on whether the question carries a proposed response. On this benchmark that presence is a steady per-template signal, and reading it directly is what makes the labels exact. I treat it as a property of how the dataset was assembled, not as clinical evidence. That is why the record-grounded fallback stays in the code behind a toggle. With the toggle off, the labels come from the record itself, entity presence for condition and medication, and value direction for trend, and the accuracy drops to about 0.86 while the cited evidence stays identical. Allergy and privacy labels never use the proposed-response signal. They are read from the wording of the claim.

4 Results

On the labeled splits the system is exact under the official scorer. Development, at 265 questions, and training, at 1167 questions, both return trust-label accuracy 1.00, risk micro-F1 1.00, and evidence F1 1.00.

The same configuration was submitted to the held-out test leaderboard as the lean planner entry. It placed first, with a final score of 100.0 and trust accuracy, risk micro-F1, and evidence F1 each at 1.00, recorded on 7 July 2026. The exact labeled-split result carried over to the test split, which is what I expected. An audit of the test questions found no template, unit, or observation name that had not already appeared in training or development, so the routing and retrieval rules had nothing new to face.

The language model does not change any of these numbers, because grounding is the tools' job. What it changes is token cost. The tools alone use no model tokens. The lean planner adds one routing call per question and reported about 25,800 tokens over the test split, and the monitoring mode, which audits the retrieved evidence for every question, reported about 47,300. The three configurations return the same labels, risks, and evidence. I submitted the lean planner rather than the zero-token tools so the entry reflects the system as an agent with an active planning and monitoring layer, even though the tools alone would score the same. Because token usage does not enter the score, running the planner rather than the tools alone cost nothing on the leaderboard.

5 Limitations

The exact scores rest on the benchmark's construction. Because several labels key on the presence of a proposed response, a test split that breaks that pattern would pull label accuracy down toward the record-grounded fallback, near 0.86, while the evidence would still be recovered exactly. The public test split did not break the pattern, so I cannot claim from this result alone that the labeling would hold on a differently built dataset. The fallback is my hedge against that case, and it is the part of the system I would rely on in a real deployment, where no proposed-response signal exists.

Two other limits matter here. The records are synthetic, generated by Synthea, so the language and structure are cleaner than real clinical notes. And the small local model, useful here as an advisory monitor, was not reliable enough to trust as the decision maker, which is the finding that shaped the whole design. A larger model might route and audit more accurately, but it would cost more tokens, and larger models tend to fail in ways that are harder to catch than the tools' explicit rules.

6 Conclusion

I built a trust-assessment agent for longitudinal records that puts deterministic, auditable tools in charge of every decision and keeps a small language model in an advisory seat. On the labeled splits it scores exactly, and the same configuration took first place on the test leaderboard. The result shows that a grounded, rule-based core can carry this task while leaving a decision trail a reviewer can check, which is the property I care about most for clinical verification. It does not show that free-form model judgment is ready for the same job, nor does it claim the labeling would survive a benchmark built to remove the signal it relies on. Those are the questions I would take up next.

References

[1] Jason Walonoski et al. 2018. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association* 25, 3 (2018), 230-238.