Large Language Models for Trauma Note Quality

Kaijie Zhang
kaz029@ucsd.edu

Viv Somani
visomani@ucsd.edu

Aaron Boussina
aboussina@health.ucsd.edu

Abstract

Code:
https://github.com/KaijieZhang0831/TQIP-Trauma-Note-LLM-Extraction

1 Introduction

1.1 Overview

An effective surgical quality program relies heavily on maintaining a comprehensive patient registry to track outcomes and benchmark against national standards. The American College of Surgeons Committee on Trauma requires trauma centers to maintain these registries for accreditation, which are critical for reporting trauma quality metrics. However, traditional registries are labor-intensive and costly. Artificial intelligence, particularly Large Language Models (LLMs), offers a potential solution to streamline this process. We hypothesized that a LLM could be applied to review patient charts and identify complications as defined by the Trauma Quality Improvement Program, offering an effective adjunct to manual chart reviews.

1.2 Prior Work

Over the last two decades trauma quality improvement programs (TQIPs), such as the American College of Surgeons (ACS) TQIP, have proven to be essential in ensuring that participating trauma centers maintain the highest standards of care. Through participation in the ACS Committee on Trauma's (COT) Verification, Review and Consultation (VRC) program, trauma centers undergo a rigorous formal assessment to verify that they are in line with all or most criteria to earn the classification of a Level I, II, or III trauma center [Resources for Optimal Care of the Injured Patient]. At the national level, this standardization improves the care of trauma patients by ensuring that institutions are well-equipped to care for patients with complex and multi-faceted injuries. Trauma centers subsequently benefit from verification and participation in this TQIP via the program's regular benchmark reports that allow them to identify deficiencies in their care processes or quality metrics relative to national rates and similarly verified institution's, allowing program leadership to implement process improvement initiatives (Hemmila et al. 2010). As a result, the ACS TQIP has left an indelible mark in the advancement of care for trauma patients at ACS verified trauma centers, with a recent study demonstrating that high performing centers (those in the lowest decile of overall risk-adjusted mortality) were more likely to be adherent to several VRC quality metrics (Cho et al. 2025).

Accurate benchmarking, quality-measure adherence, and self-assessment is dependent on the integrity of data reported from participating trauma centers (Nathens, Cryer and Fildes 2012). The ACS VRC program requires all centers, regardless of level, to maintain a trauma registry and have a written data quality plan to validate that the data being reported are high-quality. Meeting these data reporting standards can be costly, with the VRC program requiring 0.5 full time equivalents (FTEs), or registrars, per 200-300 annual patient entries that meet National Trauma Data Standard (NTDS) inclusion criteria. For most level I centers, maintenance this means employing upwards of 5.0 FTEs to keep up with patient volume, with level II centers requiring about 3.0 FTEs (Elkbuli, McKenney et al. 2020). In addition to the considerable upfront financial costs (Moore and Clark 2008), it also takes considerable time and effort to train trauma registrars to meet the requirements dictated by the ACS COT (Nathens, Cryer and Fildes 2012). ACS requires registrars to attend and pass several mandated courses. Additionally, the VRC program requires that 80 percent of patient records be completed within 60 days after patient discharge. In addition to the requirements of the ACS trauma registry, many trauma centers also participate in regional or state registry, each with their own data requirements (Hemmila et al. 2017) (Hemmila et al. 2018). These certification requirements in combination with wages and a persistently growing patient data backlogs can lead to a high rate of trauma registrar turnover (Day 2012). Although well-funded trauma centers may be able to shoulder the burden of high turnover rates among registry personnel, smaller trauma centers may find it difficult to keep up with the demands of high-quality patient data entry. These challenges are often cited as a barrier to the implementation of trauma registries in middle- and low-income countries (Bommakanti et al. 2018) (Purcell et al. 2020) (Klappenbach et al. 2024).

In an effort to facilitate data digestibility and registry-related work, many institutions have made efforts to manipulate or configure provider documentation to improve the ease of chart review and data-entry. However, this often results in additional documentation burden being placed in a field of surgery already rife with documentation issues (Ludley et al. 2023). Artificial intelligence (AI), and specifically large language models (LLMs), is one way trauma centers can reduce the time, effort, and cost it takes to maintain a trauma registry. LLMs have gained recent acclaim in medicine due to their ability to be trained on and pass board exams for several specialties and even show promise in diagnosis (Mahajan et al. 2025), (Alessandri-Bonetti et al. 2024). More recently, they have demonstrated their promise in the abstraction of complex hospital quality measures (Boussina et al. 2024). Based on this work, we hypothesized that LLMs could be used to streamline the identification of complications as defined by the NTDS, offering a faster and more cost-effective alternative to manual chart reviews performed by trauma registrars.

1.3 Relevant Data

Clinical data, especially trauma notes and medication orders, are protected from UCSD Health and sensitive; all of our data, the patient features, were stored in a monitored and protected environment. The Prompt with CoT were created based on the National Trauma Data Standard Data Dictionary 2025 Admission as the instruction. The LLM we used is ”us.deepseek.r1-v1:0” and the embedding we used is ”amazon.titan-embed-text-v2:0” via Amazon Bedrock.

2 Methods

2.1 Data and LLMs

We included all patient encounters from a large academic level 1 trauma center with a complication in the registry from January 1, 2023 through October 31, 2024. This was a convenience sample selected based on availability of data exported from the trauma registry. Institutional review board (IRB) approval was obtained with waiver of informed consent (#808297). We followed STROBE guidelines to ensure appropriate reporting of this research (Von Elm et al. 2007). Clinical notes were retrieved from the institution's electronic health record using the Fast Healthcare Interoperability Resources (FHIR) standard version R4 (Figure 1). We utilized the United States Deepseek-R1 LLM, coupled with Retrieval Augmented Generation to handle the extensive length of clinical notes (Guo 2025). Notes were first filtered using fuzzy matching against a list of keywords for 18 specific complications, including unplanned admission to ICU, unplanned intubation, severe sepsis, delirium, pressure ulcer, stroke, alcohol withdrawal, cardiac arrest with cardiopulmonary resuscitation (CPR), deep venous thromboembolisms (DVTs), acute kidney injury (AKI), unplanned visits to the operating room (OR), pulmonary embolisms (PE), catheter-associated urinary tract infections (CAUTI), myocardial infarctions (MI), ventilator-associated pneumonia (VAP), acute respiratory distress syndrome (ARDS), osteomyelitis, and superficial surgical site infection. The resulting notes were then chunked into segments and converted into vectors using the Amazon Titan Embed Text v2 model (Amazon Web Services 2024). RAG was performed using maximal marginal relevance (MMR) similarity search between the embeddings and the prompt and selecting the top 12 most similar chunks into the final prompt Carbonell and Goldstein (1998). MMR was chosen to capture diverse segments of text from highly redundant clinical notes. The abstraction guidelines from the 2024/2025 NTDS Data Dictionary informed the design of prompts for the LLM to identify each of the 18 complications. Chain-of-thoughts (only as instruction for the prompt for now) and few-shot prompting techniques were utilized. We scaled test-time compute with self-consistency decoding using minority voting (where we actually use k-of-n voting for now) (Snell et al. 2024). Specifically, inference was performed 5 times for each prompt with a temperature of 0.3 and a complication was considered present if the LLM identified the complication in at least two of the responses. The LLM also provided rationale behind the findings (from chain-of-thoughts, but not strictly defined yet), citing specific texts in the electronic health record (EHR). Better Chain-of-thoughts prompting will be applied.

2.2 Statistical Analysis

Our primary outcome was agreement between the LLM and manual reviews performed by the institution's trauma registry. We assessed sensitivity, negative predictive value (NPV), positive predictive value (PPV), and frequency of complications identified by the LLM but not the registrar. Additionally, a subset of cases and output from the LLM was reviewed and validated by clinical subject matter experts (SMEs) via manual chart review. Cases in which the LLM missed complications that the human registrars identified were specifically included in this subset. Rationale provided by the LLM facilitated targeted reviews and verification of output as true or false. Statistical analysis was performed in Python version 3.9.11.

3 Results

3.1 Performance

As a simple smoke test for AWS Bedrock with new embedding and LLMs, here is the pre-Embedding and pre-LLM optimization (raw) of experiment on 20 patient features (During extended runs we encountered several technical issues that required additional time to diagnose and stabilize, therefore we used a subset of 20 samples as a preliminary peek at the system behavior):

Table 1: Performance of the LLM-based NTDS complication extraction in 20 patient features

Complication	**Sensitivity**	**PPV**	**NPV**
Alcohol Withdrawal Syndrome	0.857	0.667	0.909
Delirium	1.000	0.667	1.000
DVT/Thrombophlebitis	0.333	1.000	0.895
Stroke/CVA	NA	0.000	1.000
Unplanned Intubation	1.000	0.200	1.000
Unplanned Admission to ICU	1.000	0.600	1.000
Severe Sepsis	NA	0.000	1.000
Pressure Ulcer	NA	0.000	1.000
Cardiac Arrest with CPR	1.000	1.000	1.000
Acute Kidney Injury	NA	0.000	1.000
Unplanned Visit to OR	NA	0.000	1.000
Pulmonary Embolism	NA	0.000	1.000
Myocardial Infarction	NA	NA	1.000
VAP	NA	0.000	1.000
ARDS	NA	NA	1.000
CAUTI	NA	0.000	1.000
Osteomyelitis	NA	NA	1.000
Superficial Incisional SSI	NA	NA	1.000
Overall Sensitivity	0.857 (18/21)		
Total TP / FP / FN / TN	18 / 40 / 3 / 299		
Average Additional Complications	200%		

Overall performance structure. Overall sensitivity is 85.71% with 18 true positives and 3 false negatives. The recall is reasonably strong and the false negative rate is relatively low. As a screening tool this leans toward the safe side. However it does not yet meet registry level standards which typically require sensitivity above 90% to 95%.

Precision structure and over prediction behavior. The average percentage of additional complications is 200%. This indicates that the number of predicted complications is far higher than the true number. The system clearly over predicts. With 18 true positives and 40 false positives the overall positive predictive value is approximately 31%. This means that about 70% of predicted positive cases are incorrect. For registry workflow this would create substantial manual review burden, increase trust cost, and reduce the likelihood of hospital adoption.

Class level error patterns. For pressure ulcer the positive predictive value is 0/10 and sensitivity is not applicable because there are no true cases. All predictions are false positives, indicating severe hallucination behavior for this complication. For unplanned intubation sensitivity is 1.0 but positive predictive value is 0.2. The model predicted five cases but only one was correct, showing a recall driven over trigger pattern. For DVT sensitivity is 0.33 and positive predictive value is 1.0. The model rarely predicts this complication but misses true cases, showing the opposite imbalance. These patterns indicate that triggering logic is highly inconsistent across complications.

Although we reused parameter settings from previous pipelines, which may not be well aligned with the new embedding model and LLM configuration, the current performance is clearly unsatisfactory. The model is recall oriented but suffers from precision collapse due to systematic over triggering and loose definition alignment. The main risk is not false negatives but definition hallucination and threshold miscalibration. In its current form the system is not suitable for deployment without significant false positive control and stricter definition enforcement. We will quickly investigate the underlying causes and adjust the configuration to bring the performance back to a reasonable and stable range.

3.2 Time Analysis

Table 2: Runtime Breakdown of the Preliminary Baseline Pipeline (5 samples)

Module	**Total Time (sec)**	**Mean Time**	**% of Total Runtime**
LLM Engine	690.08	12.32 / call	79.7%
Vectorstore Build (Embedding)	166.67	3.09 / build	19.3%
Retriever Invoke	7.56	0.14 / query	0.9%
Chunk Filtering	0.52	0.006 / call	0.06%
Text Splitting	0.015	0.003 / case	0.002%
Other Overhead	0.40	–	0.05%
Total Runtime (5 cases)	865.25 sec (100%)		

LLM inference dominates overall latency (80%), followed by embedding and vectorstore construction (19%). All preprocessing and chunk operations contribute negligibly to total runtime.

The runtime distribution is consistent with our expectations. The primary bottleneck comes from the LLM inference stage, followed by vector embedding and vectorstore construction. We observe that the current runtime is significantly slower than our previous local LLaMA based experiments, which may be due to additional latency introduced by Bedrock or the inherently longer inference time of the DeepSeek model. However, the increased runtime does not translate into better performance. Therefore, our immediate optimization focus will be on LLM inference, including parameter tuning and a detailed investigation of its behavior.

4 Discussion

Before proposing further changes, we observed that the current Amazon Bedrock setup using a new LLM and new embeddings performs substantially worse than our prior local deployment. We therefore need to diagnose the root causes and narrow the gap as much as possible. Potential explanations include mismatches between the codebase and the version described in our report, the need to re-tune model and decoding parameters, or weaker clinical vocabulary coverage in the new embedding model compared with the medically tuned hkunlp Instructor embeddings we used previously. We plan to complete this investigation by Week 7 and then move quickly to the next step.

Regarding sample size, we have roughly 500 cases available. However, given time and cost constraints, our experiments will continue to use a fixed set of about 50 cases as a consistent benchmark for comparison. After we finalize all methodologies, we will scale up and run the pipeline on the entire dataset to obtain the best final results, including ablation tests to quantify the impact of each component.

In addition, we discussed some other potential directions for improvement:

Hybrid chunking: We plan to move from “concatenate everything then split” to a hybrid strategy that respects note boundaries. We will first treat each clinical note as a unit, keeping metadata such as note type, day/timestamp, and authoring service when available. Within each note, we will apply a fixed chunk size and overlap to maintain stable lengths for embeddings and to avoid token explosions. This approach should reduce false positives caused by losing local negation context, such as “no evidence of PE” being separated from its qualifier. At the same time, it should increase recall by ensuring small but decisive phrases are not diluted inside very long combined texts. We expect this hybrid design to improve both retrieval quality and interpretability because retrieved evidence can be traced back to a specific note and section.

Chapter-by-type for speed: Notice that some note types are the main source of specific complication extractions, we are considering restructuring retrieval to mimic how registrars “flip through” a chart by note category. We will create chapters based on note type, such as ICU progress notes, discharge summaries, radiology reports, operative/procedure notes, consult notes, and ED documentation. For each NTDS complication, we will define a preferred set of chapters that are most likely to contain definitive evidence, for example ICU and radiology for ARDS or PE, and OR plus discharge for unplanned return to the operating room. Retrieval will then run primarily within these chapters, either by metadata filtering or by weighting results from preferred chapters more heavily. This reduces the candidate search space and can significantly speed up embedding search while improving precision. It also makes failure modes easier to debug because we can see whether a complication was missed due to looking in the wrong chapter.

Post-vote follow-up: We plan to keep self-consistency voting across multiple LLM runs, but add an additional "clarification" step when the vote is uncertain. If outputs disagree, confidence is low, or the retrieved evidence contains hedging language like "rule out" or "concern for," we will issue a targeted follow-up prompt. The follow-up will ask the model to cite one explicit supporting span from the retrieved text and then output a final binary decision under the NTDS definition. Overall, we expect this step to reduce false positives while preserving sensitivity for true complications.

5 Conclusion

5.1 Summary and Future Plan

Overall, our progress is still slightly behind the original timeline, but we have completed the most difficult portion of the work. Next, our direction is clear. We will first optimize the current LLM setup and align its parameters with our prior LLaMA-based experiments to achieve comparable performance. We will then run ablation studies by swapping in the new components we proposed, so we can isolate their effects and draw concrete conclusions. Finally, we will build a project website to support a clear and polished Capstone demonstration.

5.2 Contributions

Kaijie Zhang: I completed setting up the codebase and updated the pipeline to run the first end-to-end experiment. I also refactored the experiment logging and evaluation code. In addition, I proposed several follow-up improvements and am actively developing them.

Viv Somani: I completed sections 1 and 2 of the report. I also assisted in setting up the codebase.

References

Alessandri-Bonetti, Mario, Hilary Y Liu, James M Donovan, Jenny A Ziembicki, and Francesco M Egro. 2024. "A comparative analysis of ChatGPT, ChatGPT-4, and Google Bard performances at the Advanced Burn Life Support exam." *Journal of Burn Care & Research* 45 (4): 945–948

Amazon Web Services. 2024. "Amazon Titan Models: Titan Embed Text v2." https://aws.amazon.com/bedrock/titan/

Bommakanti, Krishna, Isabelle Feldhaus, Girish Motwani, Rochelle A Dicker, and Catherine Juillard. 2018. "Trauma registry implementation in low-and middle-income countries: challenges and opportunities." *Journal of surgical research* 223 : 72–86

Boussina, Aaron, Rishivardhan Krishnamoorthy, Kimberly Quintero, Shreyansh Joshi, Gabriel Wardi, Hayden Pour, Nicholas Hilbert, Atul Malhotra, Michael Hogarth, Amy M Sitapati et al. 2024. “Large language models for more efficient reporting of hospital quality measures.” *Nejm ai* 1 (11), p. AIcs2400420

Carbonell, Jaime, and Jade Goldstein. 1998. “The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries.” In *SIGIR ’98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Cho, Nam Yong, Jeff Choi, Saad Mallick, Galinos Barmparas, David Machado-Aranda, Areti Tillou, Daniel Margulies, Peyman Benharash, Academic Trauma Research Consortium et al. 2025. “Beyond American college of surgeons verification: quality metrics associated with high performance at level I and II trauma centers.” *Journal of the American College of Surgeons* 240 (2): 190–200

Elkbuli, Adel, Mark McKenney et al. 2020. “Trauma Registry Staffing and Costs: A National Survey of Level I and Level II Trauma Centers.” *Annals of Surgery*

Hemmila, Mark R et al. 2017. “Association of Hospital Participation in a Regional Trauma Quality Improvement Collaborative With Patient Outcomes.” *JAMA Surgery* 152 (8): 743–753

Hemmila, Mark R et al. 2018. “Collaborative Quality Improvement for Trauma: The Michigan Experience.” *Journal of Trauma and Acute Care Surgery* 85 (1): 200–207

Hemmila, Mark R, Avery B Nathens, Shahid Shafi, J Forrest Calland, David E Clark, H Gill Cryer, Sandra Goble, Christopher J Hoeft, J Wayne Meredith, Melanie L Neal et al. 2010. “The Trauma Quality Improvement Program: pilot study and initial demonstration of feasibility.” *Journal of Trauma and Acute Care Surgery* 68 (2): 253–262

Klappenbach, H. et al. 2024. “Trauma Registries in Low- and Middle-Income Countries: A Systematic Review.” *Injury*

Ludley, Alistair, Andrew Ting, Dean Malik, and Naveethan Sivanadarajah. 2023. “Observational analysis of documentation burden and data duplication in trauma patient pathways at a major trauma centre.” *BMJ Open Quality* 12 (2)

Mahajan, Arnav, Andrew Tran, Esther S Tseng, John J Como, Kevin M El-Hayek, Prerna Ladha, and Vanessa P Ho. 2025. “Performance of trauma-trained large language models on surgical assessment questions: a new approach in resource identification.” *Surgery* 179 , p. 108793

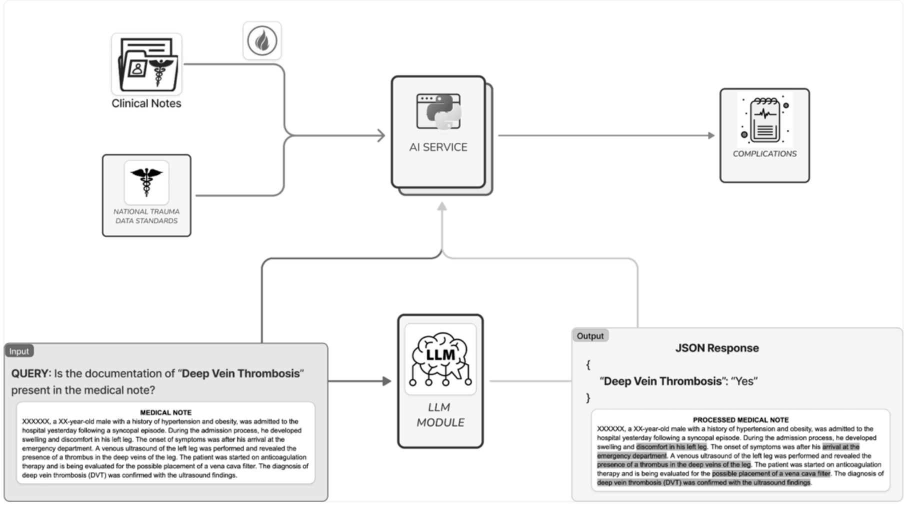
Moore, Lynne, and David E Clark. 2008. “The value of trauma registries.” *Injury* 39 (6): 686–695

Nathens, Avery B, H Gill Cryer, and John Fildes. 2012. “The American College of Surgeons trauma quality improvement program.” *Surgical Clinics* 92 (2): 441–454

Purcell, L. N. et al. 2020. “Barriers to and Facilitators of the Development of a Trauma Registry in a Low-Income Setting.” *World Journal of Surgery* 44: 4112–4119

Snell, Charlie et al. 2024. “Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters.” *arXiv preprint arXiv:2408.03314*

Von Elm, Erik, Douglas G Altman, Matthias Egger et al. 2007. “The Strengthening the Reporting of Observational Studies in Epidemiology (STROBE) Statement: Guidelines for Reporting Observational Studies.” *The Lancet* 370 (9596): 1453–1457



QUERY: Is the documentation of "**Deep Vein Thrombosis**" present in the medical note?

MEDICAL NOTE

XXXXXX, a XX-year-old male with a history of hypertension and obesity, was admitted to the hospital yesterday following a syncopal episode. During the admission process, he developed swelling and discomfort in his left leg. The onset of symptoms was after his arrival at the emergency department. A venous ultrasound of the left leg was performed and revealed the presence of a thrombus in the deep veins of the leg. The patient was started on anticoagulation therapy and is being evaluated for the possible placement of a vena cava filter. The diagnosis of deep vein thrombosis (DVT) was confirmed with the ultrasound findings.

JSON Response

```
{
  "Deep Vein Thrombosis": "Yes"
}
```

PROCESSED MEDICAL NOTE

XXXXXX, a XX-year-old male with a history of hypertension and obesity, was admitted to the hospital yesterday following a syncopal episode. During the admission process, he developed swelling and discomfort in his left leg. The onset of symptoms was after his arrival at the emergency department. A venous ultrasound of the left leg was performed and revealed the presence of a thrombus in the deep veins of the leg. The patient was started on anticoagulation therapy and is being evaluated for the possible placement of a vena cava filter. The diagnosis of deep vein thrombosis (DVT) was confirmed with the ultrasound findings.

Figure 1: Clinical notes are retrieved from the EHR via FHIR version R4. The NTDS Data Dictionary is used by the AI Service for 18 complication-specific prompts. These prompts are then used to perform RAG against the clinical notes and generate the LLM input. The LLM module processes these inputs with chain-of-thoughts, few-shot, and self-consistency prompting. The response is cast to JSON to generate the final complications output.

Appendices

A.1 Latest Project Proposal

The latest Project Proposal is attached below, which was written in first quarter.