Towards practical application of local LLMs in psychiatry

Noe Javet

June 6, 2025

Contents

Abstract

Recent advances in large language models (LLMs) have demonstrated strong performance in medical text analysis tasks, including the evaluation of clinical discharge reports. However, the high computational requirements and proprietary nature of models such as ChatGPT limit their accessibility and transparency. This thesis investigates whether small, open-source LLMs—specifically models in the 7–8 billion parameter range—can be optimized to approach the analytical quality of proprietary systems in the context of medical discharge summary evaluation. We design a task-specific evaluation framework to assess model performance across dimensions such as section completeness, clinical relevance, and reasoning fidelity. Techniques such as prompt engineering, system message adaptation, and agentic workflows are applied to enhance the output of local models. Comparative analysis with GPT-4-based baselines reveals that, while small models exhibit limitations in reasoning depth and robustness, targeted optimization significantly narrows the performance gap. Our results highlight the viability of deploying resource-efficient LLMs for structured medical report analysis under constrained environments.

1 Introduction

Large language models (LLMs) have shown remarkable capabilities in understanding, generating, and evaluating natural language, with applications spanning education, law, healthcare, and scientific research.[1] In the clinical domain, LLMs have demonstrated potential in assisting with tasks such as summarization, information extraction, and documentation analysis. Among these, the evaluation of medical discharge reports—a critical component of clinical documentation—remains a challenging problem due to the need for domain-specific reasoning, completeness verification, and structured section interpretation. Proprietary models such as OpenAI's ChatGPT have set a high bar for performance in these tasks, benefitting from instruction tuning, extensive pretraining, and scalable infrastructure. However, their deployment is limited by computational cost, lack of transparency, and privacy concerns—especially in sensitive fields such as healthcare. In contrast, open-source LLMs in the 7–8 billion parameter range offer local deployability, but their capabilities in clinical evaluation tasks remain underexplored.

This thesis addresses the following research question:

> Can small, open-source language models (8B parameters) be optimized to evaluate the completeness of medical discharge reports at a level comparable to that of proprietary models such as ChatGPT?

To answer this question we focus on the task of section-level completeness analysis: identifying whether critical information (e.g., diagnoses, medication, follow-up plans) is present and appropriately described. To this end, we develop an evaluation framework that incorporates reference annotations, model-guided assessments, and performance metrics such as precision, recall, and relevance. We further explore prompt engineering techniques, instruction design, and lightweight agentic workflows to enhance model behavior without fine-tuning.

2 Methods

To evaluate the capabilities of various language models (LLMs) in processing medical discharge reports, a multi-stage evaluation pipeline was implemented. The process consisted of document preprocessing, prompt engineering, model inference, and result evaluation using defined metrics.

2.1 Dataset

The dataset used for evaluation consists of seven medical discharge reports. Among them, two reports are considered complete: one includes section-level annotations and one does not.

Report	**Missing Sections**
Report 1	Medical History
Report 2	Substance use history
Report 3	Psychopharmacology
Report 4	Clinical course
Report 5	Diagnoses

Table 1: Overview of missing sections in the five incomplete discharge reports.

Report 1 and 2 do have empty sections, while Report 3 and 4 do have valid german statements, but without any information. Report 5 does have a valid diagnosis, but is not formal enough.

2.2 Evaluation Metric

To assess the performance of the language models in identifying incomplete sections within medical discharge reports, we define the following evaluation categories:

- **True Positive (TP)**: The model correctly identifies a section as incomplete when it is indeed missing or insufficient according to the reference annotation.
- **False Negative (FN)**: The model incorrectly classifies an incomplete section as complete, thereby failing to detect the omission.
- **Irrelevant (IR)**: The model output diverges from the prompt instructions, such as by hallucinating content, introducing unrelated information, or failing to perform the evaluation task.

2.3 Language Model Configuration

For all model evaluations, a deterministic generation configuration was used to promote consistency and minimize variance across runs. Specifically, we selected a low temperature (`temperature = 0.0`) along with constrained sampling parameters (`top_p = 0.4`, `top_k = 8`). These settings reduce randomness in token selection and encourage the model to produce concise, focused outputs aligned with the most probable completions. The rationale behind this choice is twofold. First, clinical report evaluation is a task

that prioritizes factual precision and structural adherence over creativity or linguistic diversity. Second, high-temperature outputs tend to increase hallucination rates and irrelevance, which directly impacts the reliability of model judgments in a safety-critical domain such as medicine. By enforcing a narrower decoding distribution, we ensure that the model's responses remain stable and reproducible, which is essential for systematic error analysis and metric comparison across different prompt configurations. Each model was invoked with a system message tailored to the specific evaluation task. For instance, when assessing section completeness, the system message defined what constitutes a complete diagnosis section, how to handle missing or ambiguous information, and what format the response should follow (e.g., JSON structure indicating completeness).

By default, Ollama enforces a maximum context length of 2024 or 4096 tokens, which proved insufficient for full-length medical discharge reports. As a result, a custom `Modelfile` was created for each model to explicitly raise the `num_ctx` parameter. This allowed the models to process entire documents without truncation, ensuring accurate and contextually informed evaluations.

2.4 Evaluation

The following language models were evaluated in this study:

Model	**Parameter Size**
Dolphin-LLaMA 3.1	8B
LLaMA 3.1	8B
LLaMA 3.1 Instruct (Quantized, q8_0)	8B
Mistral Instruct	7B
Hermes 3 (LLaMA 3.1 base)	8B

Table 2: Evaluated local language models and their parameter sizes.

Initially, raw discharge summaries in DOCX format were parsed using the `python-docx` library. This stage focused on extracting plain text content from the original documents while preserving section headers and structural elements. The extracted text was then sent to the LLMs using the following system prompt:

```
Du bist ein Evaluator für medizinische Austrittsberichte einer Psychiatrie.
Prüfe den Austrittsbericht auf fehlende medizinische Informationen und gib
**ein reines JSON-Objekt** nach folgendem Schema aus:

* ‘"Abschnittsname":
0‘ wenn es Evidenz gibt, dass der Abschnitt medizinisch vollständig ist
* ‘"Abschnittsname":
1‘ wenn nicht

Gib das JSON-Objekt OHNE Kommentare, Fließtext oder Einleitung zurück.
```

Subsequently, to reduce variability introduced by inconsistent formatting, the documents were manually reformatted to adhere to a markdown-like structure. These manually structured texts were then re-evaluated using the same baseline system prompt to measure improvements attributable solely to input clarity.

In the final stage, the same markdown-formatted texts were reprocessed using an agentic workflow with access to a tool.

3 Results

3.1 Evaluation Setups

Setup A: Baseline prompt applied to DOCX reports converted to plain text using `python-docx`. **Setup B**: Baseline prompt applied to manually formatted markdown versions of the same reports. **Setup C**: Agentic prompting framework that invokes tool-calling for diagnosis section analysis.

3.2 Quantitative Metrics

The best possible outcome is 5 points for true positives, 0 points for false negatives and 0 points for irrelevant.

Figure 1 shows the distribution of evaluation outcomes (TP, FN, IR) for setup A

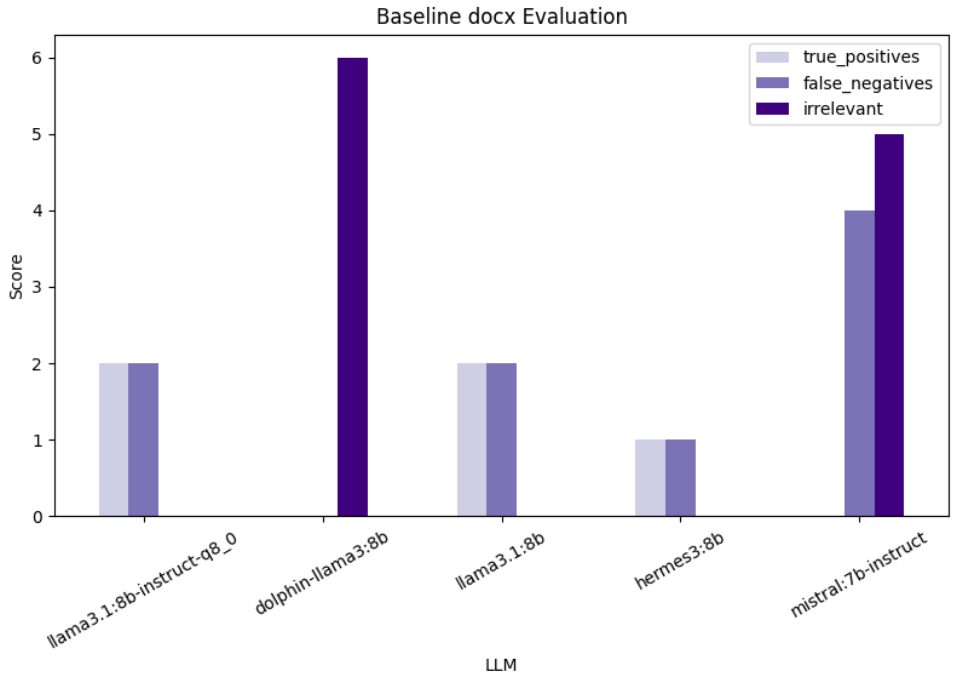

Figure 1: Evaluation outcome counts for setup A.

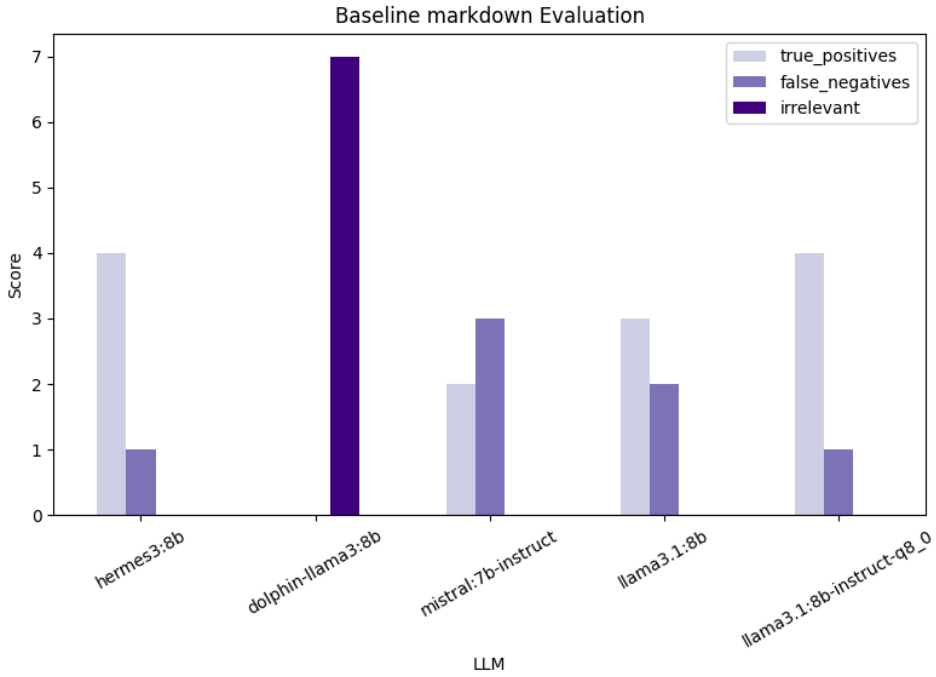

Figure 2: Evaluation outcome counts for setup B.

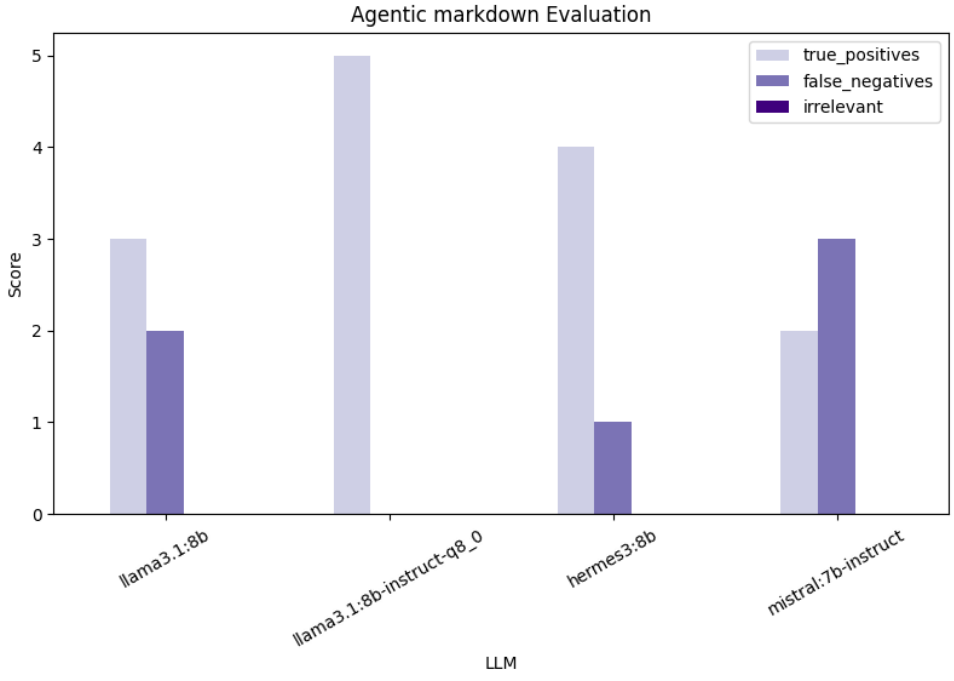

Figure 3: Evaluation outcome counts for setup C.

4 Discussion

The results highlight a major challenge in the preprocessing pipeline: the correct formatting of input documents. When using `python-docx`, footers are not reliably removed,

and paragraph boundaries are inconsistently extracted. This significantly impacts model performance—inputs derived directly from DOCX files led to noticeably worse results compared to manually formatted markdown documents. These findings underscore the critical importance of structured, well-formed input when working with language models in clinical document analysis.[2] Further experiments revealed that even minor variations in the system prompt—such as the presence or absence of a comma or blank line—can alter the model's output. This sensitivity suggests that prompt design and input consistency must be treated as part of the optimization process.

Another complicating factor is the lack of a universally accepted definition of what constitutes a "complete" medical discharge report. Standards may vary not only across countries but even between institutions, making generalization difficult.

In this work, the agentic evaluation setup focused exclusively on the diagnosis section, as this was not reliably detected as incomplete in the baseline setup—even when using a properly formatted document. In principle, the same methodology could be applied to other sections, though this would further increase computational cost and latency.

One potential direction for future work is the integration of a retrieval-augmented generation (RAG) component. This would allow individual clinics to provide their own reporting conventions and medical standards, enabling more context-aware evaluations. Combined with fine-tuning on high-quality institution-specific data, this approach could not only improve section completeness detection but also allow the model to adapt its tone and terminology to local clinical norms.

5 Conclustion

This thesis investigated whether small, open-source language models can be optimized to evaluate the completeness of medical discharge reports in a manner comparable to proprietary systems such as ChatGPT. The results show that such models—particularly those in the 7–8B parameter range—can be effectively adapted to perform section-level completeness evaluation, especially when guided by carefully designed prompts and lightweight agentic workflows.

However, this performance was demonstrated under a simplified evaluation framework based on binary classification (complete vs. incomplete) for individual report sections. While the approach enables clear metric comparison and model alignment, it does not yet capture the full nuance of clinical document quality, such as partial completeness, semantic adequacy, or medical relevance.

Future work should extend this framework to more complex classification schemes and richer annotation layers, enabling models not only to detect missing content but also to assess clinical plausibility, redundancy, or contradiction. Nonetheless, the findings presented here highlight the potential of small, local LLMs in supporting structured clinical documentation analysis—particularly in privacy-sensitive or resource-constrained environments.

6 Appendix

- Project repository on GitHub

References

[1] "Large language models can support generation of standardized discharge summaries – A retrospective study utilizing ChatGPT-4 and electronic health records". In: *International Journal of Medical Informatics* (2024).

[2] "Does Prompt Formatting Have Any Impact on LLM Performance?" In: *arXiv:2411.10541v1* (2024).