Towards practical application of local LLMs in psychiatry

Noe Javet

June 6, 2025

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. 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 investigates whether small open-source language models can be optimized to perform medical discharge report evaluation at a level approaching that of proprietary models. 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. By comparing various small LLMs to a GPT-4 baseline, this work aims to quantify the performance gap and evaluate the practical potential of deploying local models for medical documentation support.

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.

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 nine different LLMs using a uniform system prompt designed to assess completeness and relevance of medical sections.

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 a refined system prompt with more explicit instructions aimed at improving model focus and output consistency. This allowed for a comparative assessment of prompt sensitivity across models.

The evaluation metrics comprised both structural and content-based indicators: true positives, true negatives, false positives, and false negatives regarding section detection;

as well as binary assessments of section relevance and conciseness (`0/1` scale). Sections marked as missing or irrelevant were explicitly recorded to aid in qualitative analysis.

The evaluation pipeline consisted of the following stages:

Data Preparation A corpus of psychiatric discharge reports was converted from .docx to markdown using python-docx, with manual post-processing to standardize section headers and content layout.

Baseline Evaluation Each markdown-formatted document was evaluated using nine different LLMs under a shared system prompt. The models' outputs were compared against a human-annotated reference.

Prompt and Parameter Tuning Experiments with modified system prompts and various temperature settings were conducted to assess their impact on performance.

Agentic Workflow A second phase introduced tool use and inter-model collaboration. One model (planner) orchestrated evaluation plans, delegating judgment and justification tasks to others. This setup aimed to simulate a form of self-verifying reasoning.

Evaluation Metrics We measured:

True/False Positives and Negatives (per section)

Irrelevant response ratio (0/1)

Conciseness of justification (0/1)

3 Results

We evaluated the performance of small language models across three different prompting configurations on a dataset of medical discharge reports. The goal was to identify whether specific sections (e.g., diagnoses) were complete, missing, or improperly inferred. Each setup was tested on a standardized set of reports, and outputs were manually annotated against reference ground truth to classify model responses into true positives (TP), false negatives (FN), and irrelevant completions (IR).

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

Figure **??** shows the distribution of evaluation outcomes (TP, FN, IR) across the three setups. Setup C significantly reduces false negatives while maintaining a low rate of irrelevant completions, indicating improved alignment with the prompt under tool-assisted reasoning.

3.3 Section-Level Comparison

To evaluate performance on a per-section basis, we analyzed accuracy for the diagnosis section in detail, comparing the precision of model outputs in detecting missing content. Figure **??** illustrates per-model precision.

Figure 1: Evaluation outcome counts for each setup.

Figure 2: Precision for detecting incomplete diagnosis sections.

3.4 Prompt Adherence and Hallucination Rate

We also measured the tendency of each model to diverge from prompt instructions, such as adding non-existent information or hallucinating sections. Figure **??** visualizes the rate of irrelevant completions per setup.

irrelevant_completions.png

Figure 3: Percentage of completions classified as irrelevant.