When the Domain Expert Has No Time and the LLM Developer Has No Clinical Expertise: Real-World Lessons from LLM Co-Design in a Safety-Net Hospital

Avni Kothari[1, 2], Patrick Vossler[1, 2], Jean Digitale[1], Mohammad Forouzannia[1], Elise Rosenberg[2], Michele Lee[2], Jenneé Bryant[2], Melanie Molina[1, 2], James Marks[1, 2], Lucas Zier[1, 2], Jean Feng[1, 2*]

[1]University of California, San Francisco
[2]Zuckerberg San Francisco General Hospital

Abstract

Large language models (LLMs) have the potential to address social and behavioral determinants of health by transforming labor intensive workflows in resource-constrained settings. Creating LLM-based applications that serve the needs of underserved communities requires a deep understanding of their local context, but it is often the case that neither LLMs nor their developers possess this local expertise, and the experts in these communities often face severe time/resource constraints. This creates a disconnect: how can one engage in meaningful co-design of an LLM-based application for an under-resourced community when the communication channel between the LLM developer and domain expert is constrained? We explored this question through a real-world case study, in which our data science team sought to partner with social workers at a safety net hospital to build an LLM application that summarizes patients' social needs. Whereas prior works focus on the challenge of prompt tuning, we found that the most critical challenge in this setting is the careful and precise specification of *what* information to surface to providers so that the LLM application is accurate, comprehensive, and verifiable. Here we present a novel co-design framework for settings with limited access to domain experts, in which the summary generation task is first decomposed into individually-optimizable attributes and then each attribute is efficiently refined and validated through a multi-tier cascading approach.

Code — https://github.com/jjfenglab/social-wayfinder
Extended Version — https://arxiv.org/abs/2508.08504

Introduction

Safety-net programs deliver essential medical and social services to underserved communities, yet they often operate under severe resource shortages and staffing constraints. Recent advances in LLMs offer tremendous promise for amplifying services provided by these programs, but realizing that potential requires genuine participatory design. Without domain expertise and stakeholder input, AI solutions are at a substantially higher risk of perpetuating biases, being misaligned, or lacking practical utility (De-Arteaga, Fogliato, and Chouldechova 2020; Buolamwini and Gebru 2018; Shneiderman 2020). However, unlike standard software engineering, co-designing an LLM application requires overcoming the "gulf of envisionment" (Subramonyam et al. 2024): there is often a large gap between a human's initial intentions to use an LLM and their final crystallized intentions that are translatable into effective LLM prompts. Crossing this gulf can often be a long process, because the exact intentions of a project as well as an LLM's ability to produce the desired outputs are typically unclear in the initial stages.

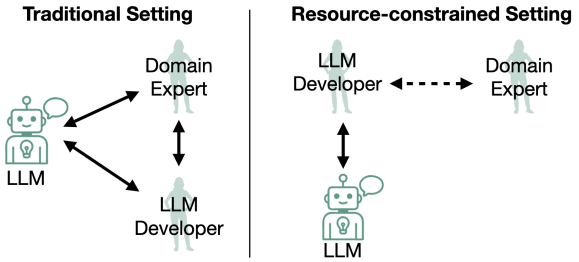

Figure 1: In traditional settings, both the LLM developer and domain expert have frequent communication to jointly iterate on and create an LLM application (solid lines mean full access). By contrast, in resource-constrained settings, the LLM developer has limited access (dashed lines) to the domain expert, which can be a major barrier to developing a truly useful application.

Traditional approaches to LLM application co-design require significant time investment from both LLM developers and domain experts, as they assume the two parties will work closely to iteratively refine their intentions and prompts (Karayanni et al. 2024). However, this overlooks a major catch-22 when developing AI for resource-constrained settings (Fig 1): the very communities that could benefit from AI assistance are also those whose resource constraints prevent them from fully engaging in the intensive co-design process. Without the help of domain experts, AI developers may not have the domain expertise and/or the full local context to build a truly useful application.

We examined challenges in LLM co-design in under-resourced settings through the lens of a real-world case study, in which our data science team was asked by the local safety-net hospital to build an LLM application to assist the hospital's inpatient social workers (SWs). The primary responsibility of SWs—the domain experts in this setting—

*For correspondence please contact: jean.feng@ucsf.edu

is to address social and behavioral determinants of health (SBDH) needs of patients and facilitate their safe discharge. The specific aim was to help with the "pre-chart summarization" step of the SW workflow, in which they identify a patient's SBDH across various domains, such as food, housing, and transportation, by reviewing past patient records. This step is necessary to prepare SWs for in-person discussions with the patient and to locate suitable social services. It is also extremely time-intensive, requiring manual review of fragmented information across multiple encounters in patient charts. A team of 23 inpatient SWs at the hospital collectively spends more than 300 hours per week on pre-chart summarization across their caseloads. This is problematic in a safety-net hospital that faces resource constraints and understaffing, where the number of patients who need social services exceeds the team's current capacity. LLMs have the potential to substantially accelerate the SWs' workflow by extracting and summarizing SBDH information from clinical notes.

Nevertheless, our data science team met major practical barriers in our initial attempts to use existing approaches. The chief challenge is that much of the existing LLM literature assumes the intention, i.e. the desired goals for using an LLM, are already precisely specified (Agrawal et al. 2022; Boussina et al. 2024; Subramonyam et al. 2024; Shankar et al. 2025; Sivaraman et al. 2025). However, unlike well-studied AI applications, novel uses in resource-constrained settings often lack established patterns or precedents, forcing teams to simultaneously define the problem space and develop solutions. Operating alone, neither the AI developer nor the domain expert was able to fully crystallize their intentions, as the AI developer has limited domain expertise and the resource-constrained domain expert has limited bandwidth. As such, methods such as human-driven prompt tuning (Goyal, Li, and Durrett 2022; Zhang et al. 2024; Aly, Soliman, and AbdelAziz 2025; Williams et al. 2025b,a) and auto-prompting (Arawjo et al. 2024; Kim et al. 2024; Shankar et al. 2024; He et al. 2025; Wang et al. 2024) lack sufficient utility and/or applicability in resource-constrained settings. Furthermore, existing co-design frameworks were similarly impractical, as the co-design process was too burdensome for experts from these communities (Majumdar et al. 2025). Finally, beyond intention formation, LLM validation was also challenging for the very same reasons.

Overcoming these barriers requires new design strategies. We introduce a dynamic, multi-tier framework that was significantly more effective and efficient in leveraging our resource-constrained domain experts through the following strategies (Fig 2):

- Assemble a team to close the gap: Assemble a team that fills the gap in domain expertise and availability, so that the team is not just at the extreme ends of the spectrum.
- Decomposing free-form summaries into attributes: Modularize the summarization task into semi-structured attributes that can be individually optimized using fragmented data and targeted feedback.
- Multi-tier cascade for intention formation: Iteratively refine the precise definition of each summary attribute by bootstrapping from existing organizational artifacts and then dynamically eliciting targeted, concrete feedback through a multi-tier cascade.
- Multi-tier validation: Validate the extracted summary attributes using a multi-tier cascade, in which LLM-as-a-judge serves as a scalable but imperfect guide and human experts provide higher-quality targeted feedback.

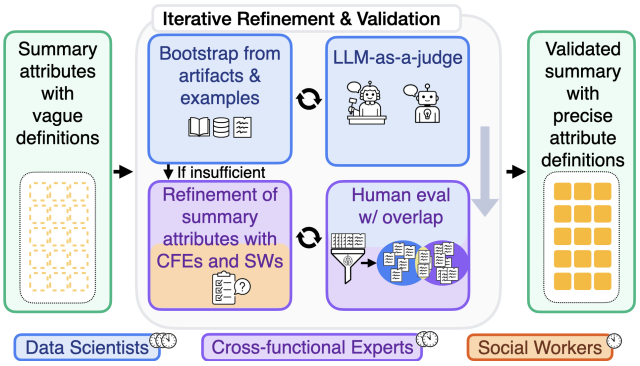

Figure 2: To efficiently co-design an LLM application starting from vaguely defined project goals, the framework recruits cross-functional experts (CFEs) to close the expertise gap, decomposes the task of generating free-form summaries into structured attributes, and iteratively refines and validates each attribute. A multi-tier cascading approach is used to minimize the number of requests sent to the domain experts who had limited availability (social workers (SWs) in this case study).

Through this approach, the first prototype of the LLM application achieved high overall accuracy and was viewed by SWs at the safety net hospital as being very likely to accelerate their workflow.

Problem Definition

Our aim was to develop an LLM-based summarization tool for SWs at our local safety-net hospital that was accurate, comprehensive, useful, and verifiable. Pre-chart summarization requires synthesizing information from multiple note types in the electronic health record (EHR), including History & Physical examinations, discharge summaries, progress notes, and various specialty consults spanning psychiatry, social work, behavioral health, care management teams across inpatient, outpatient, and emergency department contexts. Key SBDH domains to be covered included housing stability, food security, substance use, mental health, safety concerns, insurance status, immigration-related barriers, transportation access, durable medical equipment needs, and outpatient therapy arrangements.

While the task initially appeared well-scoped—synthesize social needs from EHR notes across defined domains—it presented four obstacles that are common when developing AI for under-resourced settings. Here we detail how each challenge manifested in our specific context and how it inhibited the use of existing methods:

(C1) Resource constraints and expertise gap: Despite having close collaborations with the safety net hospital, the data science team had limited expertise in social work and thus did not have the skillset needed to craft a high-quality

prompt alone. Simultaneously, the SWs, managing large caseloads, could only commit to meeting one hour once every few weeks, with occasional emails in between.

(C2) Lack of gold-standard summaries: Pre-chart summarization existed entirely as tacit knowledge. It was an undocumented process that happens primarily "in the SW's head" with no written examples, templates, or formal documentation. Without existing summaries, we could not finetune or train LLMs for this specific task. Similarly, the absence of gold-standard labeled data made automatic prompt tuning methods infeasible, as there was no consensus on what constituted a good summary.

(C3) Underspecified design requirements: Neither existing literature nor institutional materials provided templates for pre-chart summarization, particularly for safety-net populations. Beyond lacking prompting guidance, we faced a more fundamental challenge: determining which of the many details in a patient's EHR were relevant to surface in the LLM application. The specification task of defining what constituted a useful summary required domain expertise that the data science team lacked.

(C4): Evaluation: Rigorous validation of LLM extractions is necessary for safe and effective us of AI in high-stakes domains like healthcare. Due to our limited resources, a key concern was balancing the accuracy and reliability of the LLM application with resource use/cost.

Related Work

LLM summarization in healthcare: Prior works have demonstrated that LLMs have the potential to generate useful summaries of patient charts (Williams et al. 2025b,a; Bednarczyk et al. 2025; Gero et al. 2024). However, evaluating these summaries is challenging (Alaa et al. 2025), as they rely on human effort to track different attributes and facts. Closest to this work, (Gero et al. 2024) proposes structuring summaries into individual attributes to assist with LLM-as-a-judge evaluations. However, (Gero et al. 2024) uses vaguely-defined attributes from a general ontology. As such, their LLM-as-a-judge pipeline finds that the generated extractions have moderate agreement rates with the ground truth. In contrast, this work collaborates with domain experts to construct attributes that are precise and relevant to local needs, resulting in much higher agreement rates.

LLM Development with Human in the loop: Prior methods to develop LLM based applications focus on collaborating with the domain expert to iteratively co-design a product. This is done through gathering product requirements, iterating on the prompt (Reza et al. 2025; Karayanni et al. 2024; Shah 2023) and assessing the evaluation criteria through tools such as EvalLM (Kim et al. 2024), EvalGen (Shankar et al. 2024), ChainForge (Arawjo et al. 2024), and LLM Comparator (Kahng et al. 2024). These frameworks develop user interfaces for domain experts to interact with to iteratively improve the LLM system. However, these approaches do not address low-resource settings where domain experts have limited availability and would require significant time to learn the terminology and tools used in LLM applications. Our framework aimed to address the challenge of incorporating domain expertise despite this challenge.

Autoprompting: Auto-prompting methods automatically improve the prompt by providing feedback to an LLM (Shin et al. 2020; Yang et al. 2023; Zhou et al. 2022; Pryzant et al. 2023). Numerous tools such as DsPy (Khattab et al. 2023), Textgrad (Yuksekgonul et al. 2024), and Claude's prompt optimizer have been developed to help find the most optimal prompts. Generally, these tools require a ground truth dataset for optimization, but recent methods have proposed solutions for optimizing without a labeled evaluation dataset, a regime we were currently working under. These approaches structure evaluation by either asking a domain expert or DS (Arawjo et al. 2024; Kim et al. 2024; Shankar et al. 2024; He et al. 2025; Wang et al. 2024) to provide feedback on the outputs by using multiple LLM judges to critique the LLM candidate's answer (Zhu, Wang, and Wang 2023; Zheng et al. 2023). When domain experts are asked to provide feedback, they still require significant time from them to understand LLM basics, learn how to provide feedback, and annotate a large number of examples across all the prompt iterations (Szymanski et al. 2024; Karayanni et al. 2024). On the other hand, while using multiple LLM judges can reduce expert burden, this approach often fails to clarify domain-specific requirements and definitions. This has made autoprompting most successful for optimizing over a predefined evaluation metric rather than precisely defining domain-specific requirements. Our framework addressed scenarios where we had to simultaneously optimize a prompt and define product requirements.

Clinical Social Needs Extractions: Prior methods for social needs extraction from clinical notes focus on the setting of single notes that are brief and well-structured, whereas real-world settings involve complex, longitudinal notes (Yu et al. 2024; Guevara et al. 2024; Lybarger et al. 2023; Mahbub et al. 2024; Bedi et al. 2025). Furthermore, clinical notes for safety-net patients are incredibly more complex, as any single note may contain numerous references to SBDH, as well as SW jargon and abbreviations. Finally, existing methods employ generic definitions of social needs that fail to capture local contexts and practical utility in day-to-day settings, and do not address a key responsibility of clinical SWs: facilitating safe patient discharge. In contrast, our aim was to develop an LLM application that addresses the real-world needs of SWs.

Initial (Failed) Attempts

We initially tried applying existing LLM application design techniques to this problem setting, a subset of which are listed in Table 1. These initial attempts failed because existing methods were met with various barriers in this resource-constrained setting. While the attempts were individually unsuccessful, they highlighted two critical learnings that motivated our eventual design strategy.

First, it was evident that the DSs could not rely heavily on the SWs to define the AI application or guide the AI model itself. There needed to be a way to "bootstrap" the process without needing SW input (as in "pull oneself up by one's bootstraps"), so that SWs could provide meaningful feedback with the limited bandwidth they had. An initial direction was to use social worker textbooks to identify the

Approach	Challenge	Failure Modes
DSs shadow SWs due to time constraints, then independently author prompts	(C1) Resource constraints and expertise gap	DSs still lack sufficient understanding of the practical, real-world needs that SW encounter
Use previous social work consult notes as ground truth summaries	(C2) Lack of gold standard summaries	Past notes are a reflection of verbal conversations with the patient, rather than summaries of the patient record
SWs specify general requirements, DSs translate specifications into a prompt, and SWs are asked to prompt tune	(C3) Under-specified design requirements	DSs struggled to translate vague requirements into clear prompts, while SWs lacked bandwidth for iterative prompt tuning on complex summaries
Ask SWs to annotate clinical notes for relevant SBDH information	(C4) Evaluation	SWs had time to annotate a few notes at most, which is insufficient for comprehensive evaluation

Table 1: Initial attempts to develop an LLM application in resource-constrained setting using existing approaches. DS=data scientist, SW=social worker

relevant information for them (Gunasekar et al. 2023). However, there is a significant divide between theoretical social work knowledge and the practical knowledge in real-world hospital settings. Instead we needed to obtain organizational knowledge without over-burdening SWs and add new team members who could help bridge the large knowledge gap between the DSs and SWs.

Second, the goal of generating completely free-form AI-written summaries was too ill-defined. SWs found that requests for well-specified templates and examples were too time-consuming. Furthermore, free-form summaries are too difficult to evaluate objectively. Instead, we needed to break down a summary into individual attributes. Through modularization, we could individually refine each attribute's precise definition and prompt-tune in a much more targeted and efficient fashion.

Resource-efficient LLM Co-Design

To conduct LLM application co-design in this resource-constrained setting, we introduced a modular, multi-tiered approach that allows different contributors to focus on individual components and iterative refinement across tiers (Fig 2). This began with organizing a team with a sufficient range of expertise and availability (Step 1) and decomposing our complex task into manageable components (Step 2) to facilitate better collaboration and prompt refinement. The LLM application was then iteratively refined by cycling between tiered tuning of the attribute definition and prompt (Step 3) and tiered validation of the LLM extractions (Step 4). We describe the general design strategies and then how each were implemented in this case study.

Step 1: Assemble a team that closes the gap

Design strategy: Traditional LLM application development relies solely on domain experts and data scientists, which creates a bottleneck in resource-constrained settings. To fill this gap, include additional team members with cross-functional expertise who have sufficient availability to serve as the bridge and buffer between DSs and the full domain experts.

Implementation: Our team was initially composed of those at the extreme ends of the domain expertise and availability spectrum, with DSs at one end and SWs at the other. To fill the gap, we recruited cross-functional experts (CFEs) who had moderate, though not full, availability and interdisciplinary expertise spanning real-world clinical work and data science. Thus the final team consisted of SWs who have deep subject matter knowledge but were limited to only a few hours of contribution per month, CFEs who had sufficient time and context to help translate some of the SW needs to DSs, and the DSs who were the most available but lacked domain expertise. (More specifically, our CFE team members consisted of clinical care providers and clinical researchers with some data science exposure.)

By having a range in expertise and availability levels, the DSs would escalate questions to SWs only when necessary, ensuring that each group's time was used as efficiently as possible.

Step 2: Decompose the free-form summary

Design strategy: Deconstruct free-form summaries into a set of attributes to be presented in a semi-structured format within the LLM application. Modularity provides the structure to independently tune, optimize, and validate attributes. LLMs also tend to extract information for attributes more accurately than free-form summaries. The user interface can also be designed in a modular fashion.

Implementation: Based on the list of SBDH screening areas from the Centers for Medicare & Medicaid Services (CMS) (Centers for Medicare & Medicaid Services) and the day-to-day operational needs of the hospital SWs, the data science team decomposed the summarization task into 15 attributes (see final list in Appendix), thereby reframing the summarization task into an information extraction task. Although the attribute definitions were initially vague (i.e., a summary attribute may simply be defined as "housing instability"), this modularization provided the overall structure for efficient downstream intention formation and prompt creation.

Furthermore, the user interface could now also reflect this modularity, where not all elements necessarily had to be human-validated. Rather, certain elements could be labeled as "human-validated" while others may highlight a need for more detailed verification by the SW during live use by reviewing quotes from the original clinical note (Figure 3).

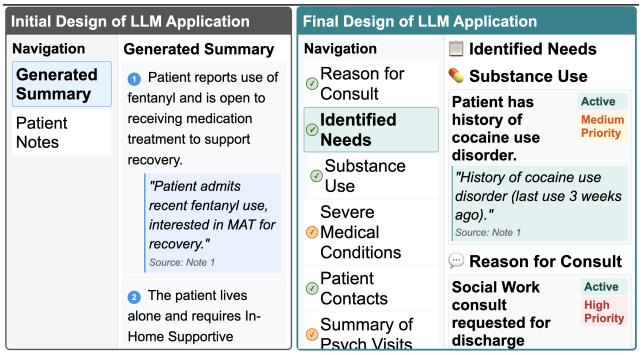

Figure 3: The initial design of the LLM pre-chart summarization application was to present a free-form summary (top). Given practical limitations in resource-constrained settings, the final design (bottom) presents a structured patient summary as 15 individual components, where components that underwent validation are highlighted by green checkmarks and attributes needing live verification are indicated by orange checkmarks. Supporting quotes from clinical notes are chosen to allow for easy verification during use of the application.

Step 3: Tiered refinement of summary attributes

Design strategy: Individually refine the attribute definitions and prompts using an iterative, multi-tier cascaded approach. To initialize the refinement process with sufficiently well-defined attributes, bootstrap the process by assembling and repurposing readily available resources. Then, moving up the domain expertise ladder, elicit concrete, data-driven feedback from CFEs and domain experts to refine attribute definitions and prompts.

Implementation: To make the most efficient use of resources, the team created a three-tier cascade. The first tier involves DSs constructing sufficiently good attribute definitions and prompts to initialize the refinement process with CFEs and SWs. The second tier involves CFEs working with DSs to check and refine attribute definitions and prompts and address as many points of confusion as possible. In the third tier, SWs provide targeted, "gold-standard" feedback. While generally the progression was from tier 1 to 3, the process was often cyclic, bouncing between tiers 1 and 2 before reaching tier 3.

Tier 1: Bootstrap from accessible organizational artifacts. The DSs found two data sources at the safety-net hospital that could be assembled in a piecemeal fashion to initialize attribute definitions and prompts that were mature enough for the iterative refinement process with limited feedback. The first was *onboarding materials used to train new SWs*, which contains terminology and resources specific to the local context. Despite the fact that onboarding documents were scattered across the organization and no single document comprehensively covered all social needs, DSs could use these to teach an LLM to tune attribute definitions and prompts; in fact, the team found this to be more effective than hand-translating the onboarding document themselves.

The second artifact was *past SW notes*. Although these notes are summaries of patient meetings and are not pre-chart summaries, these notes still provide useful example language and jargon for how different SBDH domains are discussed. Quotes from these notes can thus serve as useful in-context learning (ICL) examples. To find candidate quotes to include, DSs used an LLM to scrape notes for potentially useful quotes. CFEs could also later help select among these quotes, which ones would be most useful to include as ICL examples.

Tier 2: Gather concrete feedback by asking CFEs to extract attributes. Once the attribute definitions and prompts reached a minimum level of maturity, CFEs were able to provide concrete feedback. They were asked to annotate a small set of notes following the same instructions as the LLM, which helped highlight points of confusion and disagreements between the CFEs, DSs, and the LLM. The CFEs and DSs were then able to collaboratively improve the clarity and specificity for most attributes. Nevertheless, there still remained unresolved points of confusion that needed input from SWs.

Tier 3: Consult SWs on high value cases. For attributes where extraction instructions caused confusion or there was high annotator disagreement, we asked for explicit feedback from the SWs. This targeted approach reduced the number of attributes requiring their input and improved the speed of prompt refinement. We elicited their feedback by asking for (i) a review of the attribute definitions and prompts for completeness and alignment with their workflow and (ii) help annotating a small subset of notes that CFEs had trouble with. This led to further refinement of the summary attributes. First, annotations by the expert taught team members about the subtle distinctions, which led to further refinement of the attribute definitions and prompts. Second, certain attributes had to be updated because onboarding materials contained outdated information. Finally, the attribute of "interpersonal violence" (IPV) was found to require professional judgment that CFEs and DSs did not possess, which meant that validation of this attribute would not be feasible. As such, the team decided to broaden the attribute to encompass all "safety concerns" (e.g., anywhere from mild to more severe safety concerns). This could be much more accurately identified and validated by the entire team, thereby de-risking the LLM application.

Step 4: Tiered evaluation of LLM extractions

Design strategy: To track clarity of intermediate attribute prompts, use low-cost validation methods like LLM-as-a-judge. To validate the extractions for the final attributes with high-cost human annotations, carefully plan and power the study using methods from experimental design.

Implementation: As the SWs had no bandwidth to assist with validation of the LLM extractions, there were only two tiers from the validation step. Tier 1 involves DSs relying on LLM-as-a-judge as an approximate guide. Tier 2 is a carefully designed human validation study with both CFEs and DSs annotating notes.

Tier 1: Evaluate clarity of intermediate prompts using LLM-as-a-judge. It is helpful to measure progress across

Version	Content
Prompt1 (Base)	Review the notes and identify any mention of the social need category of Mental Health. Extract actions including previous, current, and planned actions.
Prompt2	**Added:** When extracting actions, look for interventions such as IMD facilities, LSAT, MHRC, ADU, psychiatric holds/conservatorship
Prompt3	**Refined category definition:** Mental Health - Depression, anxiety, suicidal thoughts, or psychiatric symptoms impacting functioning.
Prompt4	**Added examples:** Social needs category of "mental health" (non exhaustive list): Hallucinations, severe mood swings, etc. **Added exclusions:** Examples where this should NOT be annotated (e.g., "res was sent to ER for evaluation")

Table 2: Example of prompt tuning for a specific attribute using the tiered approach. Orange text highlights changes from the previous iteration. Each revision makes the extraction requirements more specific and clear. The transitions between prompts reflect the following: Prompt1 → Prompt2 incorporates interventions from SW onboarding documentation; Prompt2 → Prompt3 includes feedback from CFEs; Prompt3 → Prompt4 integrates fragmented examples drawn from previous SW notes.

each revision of the attribute definitions and prompts. Because human annotations at each iteration are too expensive, we use an LLM-as-a-judge pipeline (Zheng et al. 2023), in which an LLM was asked to judge the LLM application's extraction given the original patient note. Generally, we expect attribute definitions and prompts that are more vague to have more disagreements between the LLM application and the LLM-as-a-judge. So, an LLM-as-a-judge pipeline can approximately assess a prompt's clarity, with the benefit that the LLM-as-a-judge pipelines can be run at scale and minimal cost (Gu et al. 2024; Zheng et al. 2023). We refer the reader to the Appendix for an example LLM judge prompt.

Tier 2: Adaptive sampling for efficient human evaluation. For the human annotation study, we leveraged well-established strategies from experimental design to minimize human annotation costs while maintaining statistical rigor. First, we conducted a power analysis to determine the minimum sample size needed to quantify the LLM's performance with statistical confidence. Our primary goal was to establish that the LLM's sensitivity was above predefined minimum threshold, τ, which translates to testing the null hypothesis H_0 : sensitivity $< \tau$. Using standard sample size calculations, one can then determine the minimum number of notes that needed to be annotated.

Second, we used adaptive sampling to ensure that all attributes would be adequately annotated, even the ones that are rarer. However, classical approaches to adaptive sampling could not be directly applied because there were no structured data attributes to use for calculating sample weights. As such, an LLM first extracted a probability $\hat{p}_{ij}$ of attribute j appearing within note i (a value between 0 and 1 means the LLM is unsure). Then for attribute j, the probability of selecting note i for inclusion is $\max(\rho_{\min}, \hat{p}_{ij})$, where $\rho_{\min}$ (say 0.05) is a minimum selection probability.

Experimental Results

We present results from a tiered evaluation of the LLM application. Experiments were run using a PHI compliant version of GPT 4o (OpenAI, 2024) on clinical notes for patients admitted at Zuckerberg San Francisco General Hospital between January 2023 to January 2024. Patient summaries were generated using the following note types: Consults (Psychiatry, Social Work, Behavioral Health, Care Management, Complex Care Management), ED Provider Notes, Discharge Summaries, Progress Notes, and History & Physical. Summaries were based on notes from the two most recent encounters, with a maximum of 10 notes per note type.

LLM-as-a-judge Results

We first used LLM-as-a-judge to (approximately) evaluate attribute clarity across each iteration of prompt refinement. To evaluate the LLM application across patients with varying levels of medical and social needs, we sampled 99 patients stratified by note volume, resulting in a total of 1,216 notes. We tested four prompt refinements: the initial attribute-level prompt written by the DSs (*Prompt1*), the prompt after incorporating onboarding materials (*Prompt2*), then after receiving feedback from CFEs (*Prompt3*), and the final prompt that incorporated fragments from previous social work notes as ICL examples (*Prompt4*).

Table 2 shows how tiered tuning of attribute definitions and prompts incrementally increased the concordance between the LLM application and LLM-as-a-judge.[1] In this context, concordance is the rate at which the LLM judge deems the application's extraction correct. Despite limited access to domain experts, this tiered prompt tuning approach demonstrated a statistically significant improvement in almost every attribute, with an 18% overall improvement in concordance with the LLM-judge (confidence intervals and p-values are shown in the Appendix). Several attributes increased in concordance substantially—by over 25%—from initial to final prompt versions, including Outpatient Therapy and Reason for Consult. These attributes improved the most because their definitions require more specialized terminology and knowledge that had to be elicited from domain experts iteratively. Finally, whereas the initial prompt

[1]Table 3 shows 13 of the 15 attributes because two attributes were added at the end of the prompt tuning process but before human validation.

	Prompt1	*Prompt2*	*Prompt3*	*Prompt4*
Alcohol Use	0.68	0.88	0.93	0.94*
Medical Equipment	0.91	0.97	0.94	0.99*
Food Insecurity	1.00	0.98	0.98	0.99
Housing	0.75	0.77	0.94	0.93*
Immigration	0.94	0.97	0.99	1.00*
Mental Health	0.80	0.88	0.93	0.95*
Patient Contacts	0.78	0.85	0.98	0.99*
Safety	0.85	0.92	0.84	0.91
Reason for Consult	0.54	0.86	0.88	0.92*
Substance Use	0.73	0.89	0.90	0.93*
Opiod Use	0.82	0.76	0.87	0.88
Outpatient Therapy	0.51	0.72	0.65	0.90*
Tobacco Use	0.70	0.93	0.96	0.96*
Overall/Average	**0.77**	**0.88**	**0.91**	**0.95***

Table 3: Concordance between the LLM application and the LLM-as-a-judge across the patient summary attributes. Results are shown for four prompts that were iteratively refined using tiered attribute specification and prompt tuning approaches (ordered left to right). Stars denote a statistically significant increase ($p < 0.05$) from *Prompt1* to *Prompt4*.

exhibited high variability in LLM-judge concordance across the different attributes, the final prompt exhibits *much* lower variability and thus better quality control.

Human Validation Results

After the prompt refinement process, we compared model extractions with human annotations for the final prompt (*Prompt4*). We asked four CFEs and DSs to annotate notes. Prior to annotation, the four annotators underwent training through three iterative sessions designed to address any ambiguities in attribute definitions. Between sessions, we assessed inter-annotator agreement to identify areas requiring clarification and engaged SWs to annotate complex social needs cases for additional annotator training. The final human annotation study involved 85 notes, selected through stratified random sampling (in expectation, each attribute should appear at least five times in the dataset). To assess inter-annotator reliability, we assigned 10% of notes to multiple annotators.

Table 4 presents three measures: the rate at which the two annotators' extractions match (`inter-ann.`); the rate at which the LLM application's extractions match human annotations (`LLM app`); and the rate at which the LLM judge assessments align with human annotators judgements (`LLM judge`). The Appendix additionally includes Gwet's Agreement Coefficient 1, a chance-corrected agreement measure (Philip et al. 2025).

The LLM application achieved an overall agreement rate with human annotators of 0.89, which is the same as the overall agreement rate between annotators. This illustrates the potential of the LLM application to significantly improve the SW workflow. Furthermore, the overall agreement between the LLM-as-a-judge and annotators was even higher, achieving 0.92. This illustrates how LLM judges are generally helpful in measuring progress in prompt clarity.

If we further break down the results by attribute, we found that the agreement rate between LLM extractions and annotators exceeded 0.8 for nearly all attributes except for two:

	Inter-ann.	LLM app	LLM judge
Alcohol Use	0.94	0.98	0.88
Medical Equipment	1.00	0.93	0.94
Food Insecurity	1.00	0.95	0.98
Housing	0.88	0.85	0.93
Immigration	1.00	0.98	1.00
Mental Health	0.63	0.81	0.88
Patient Contacts	0.88	0.72	0.88
Safety	0.81	0.86	0.92
Reason for Consult	0.81	0.65	0.79
Substance Use	0.94	0.88	0.88
Opiod Use	0.93	0.98	0.99
Outpatient Therapy	0.88	0.88	0.86
Tobacco Use	0.88	0.95	0.91
Activities of Daily Living	0.88	0.95	0.94
Admission Reason	0.88	0.84	0.85
Overall	**0.89**	**0.89**	**0.92**

Table 4: Agreement rates when extracting social need attributes between annotators, between the LLM application and annotators, and between annotators and the LLM-as-a-judge.

"Patient Contacts" and "Reason for Consult." These two attributes had notably lower agreement rates, even though the LLM-as-a-judge had scored their LLM extractions highly. This shows that while high LLM-judge scores generally imply that a prompt is sufficiently clear and accurate, it is not always the case. Indeed, the LLM-as-a-judge had some of the lowest agreement rates with the human annotator for these two attributes (0.88 and 0.79, respectively). Thus, LLM-as-a-judge can be useful as a general guide but should not be treated as gold-standard. *Human feedback and validation are still critical to have comprehensive quality control.*

Discussion

Co-design of LLM applications for resource-constrained settings introduces novel challenges. There is often a wide gap between the AI developer and the domain expert's initial intentions to build an application and the final creation of such an application. This gap involves numerous challenges, including iteratively clarifying the team's intentions and what exact information should be extracted, refining the corresponding prompts, and validating the final results. The primary goal of this work is to highlight the need for new co-design strategies, as existing methods are too intensive for use in low-resource settings. This manuscript presented a new multi-tier cascading framework that facilitates efficient, collaborative LLM application development. Validation results demonstrate that this iterative methodology improved self-consistency across prompt iterations and high agreement rates with human annotators. While this multi-tier framework was only demonstrated in this specific use case, we anticipate this framework, or at least its basic design strategies, could be useful in other under-resourced settings as well.

This study was conducted with IRB approval.

Acknowledgements

The PROSPECT lab (JF, LZ, AK, PV, JM) thanks Zuckerberg Priscilla Chan quality improvement fund via the San Francisco General Foundation for funding this project. We want to thank the entire Zuckerberg San Francisco General Hospital Social Work Team for their invaluable feedback.

References

Agrawal, M.; Hegselmann, S.; Lang, H.; Kim, Y.; and Sontag, D. 2022. Large language models are few-shot clinical information extractors. *arXiv preprint arXiv:2205.12689*.

Alaa, A.; Hartvigsen, T.; Golchini, N.; Dutta, S.; Dean, F.; Raji, I. D.; and Zack, T. 2025. Position: Medical Large Language Model Benchmarks Should Prioritize Construct Validity. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Aly, W. M.; Soliman, T. H. A.; and AbdelAziz, A. M. 2025. An Evaluation of Large Language Models on Text Summarization Tasks Using Prompt Engineering Techniques. *arXiv preprint arXiv:2507.05123*.

Arawjo, I.; Swoopes, C.; Vaithilingam, P.; Wattenberg, M.; and Glassman, E. L. 2024. ChainForge: A Visual Toolkit for Prompt Engineering and LLM Hypothesis Testing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, 1–18. ACM.

Bedi, S.; Liu, Y.; Orr-Ewing, L.; Dash, D.; Koyejo, S.; Callahan, A.; Fries, J. A.; Wornow, M.; Swaminathan, A.; Lehmann, L. S.; Hong, H. J.; Kashyap, M.; Chaurasia, A. R.; Shah, N. R.; Singh, K.; Tazbaz, T.; Milstein, A.; Pfeffer, M. A.; and Shah, N. H. 2025. Testing and evaluation of health care applications of large language models: A systematic review. *JAMA*, 333(4): 319–328.

Bednarczyk, L.; Reichenpfader, D.; Gaudet-Blavignac, C.; Ette, A. K.; Zaghir, J.; Zheng, Y.; Bensahla, A.; Bjelogrlic, M.; and Lovis, C. 2025. Scientific evidence for clinical text summarization using large language models: scoping review. *Journal of Medical Internet Research*, 27: e68998.

Boussina, A.; Krishnamoorthy, R.; Quintero, K.; Joshi, S.; Wardi, G.; Pour, H.; Hilbert, N.; Malhotra, A.; Hogarth, M.; Sitapati, A. M.; et al. 2024. Large language models for more efficient reporting of hospital quality measures. *Nejm ai*, 1(11): AIcs2400420.

Buolamwini, J.; and Gebru, T. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, 77–91. PMLR.

Centers for Medicare & Medicaid Services. 2024. SDOH screening. https://www.cms.gov/priorities/innovation/files/worksheets/ahcm-screeningtool.pdf. Accessed: 2025-07-28.

De-Arteaga, M.; Fogliato, R.; and Chouldechova, A. 2020. A case for humans-in-the-loop: Decisions in the presence of erroneous algorithmic scores. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, 1–12.

Gero, Z.; Singh, C.; Xie, Y.; Zhang, S.; Subramanian, P.; Vozila, P.; Naumann, T.; Gao, J.; and Poon, H. 2024. Attribute Structuring improves LLM-based evaluation of clinical text summaries. *arXiv [cs.CL]*.

Goyal, T.; Li, J. J.; and Durrett, G. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Guevara, M.; Chen, S.; Thomas, S.; Chaunzwa, T. L.; Franco, I.; Kann, B. H.; Moningi, S.; Qian, J. M.; Goldstein, M.; Harper, S.; Aerts, H. J. W. L.; Catalano, P. J.; Savova, G. K.; Mak, R. H.; and Bitterman, D. S. 2024. Large language models to identify social determinants of health in electronic health records. *npj Digital Medicine*, 7(1): 1–14.

Gunasekar, S.; Zhang, Y.; Aneja, J.; Mendes, C. C. T.; Del Giorno, A.; Gopi, S.; Javaheripi, M.; Kauffmann, P.; de Rosa, G.; Saarikivi, O.; et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

He, H.; Liu, Q.; Xu, L.; Shivade, C.; Zhang, Y.; Srinivasan, S.; and Kirchhoff, K. 2025. CriSPO: Multi-aspect critique-suggestion-guided automatic prompt optimization for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 24014–24022.

Kahng, M.; Tenney, I.; Pushkarna, M.; Liu, M. X.; Wexler, J.; Reif, E.; Kallarackal, K.; Chang, M.; Terry, M.; and Dixon, L. 2024. LLM Comparator: Visual Analytics for Side-by-Side Evaluation of Large Language Models. arXiv:2402.10524.

Karayanni, N.; Awwad, A.; Hsiao, C.-L.; and Shanmugam, S. P. 2024. Keeping experts in the loop: Expert-guided optimization for clinical data classification using large language models. *arXiv preprint arXiv:2412.02173*.

Khattab, O.; Singhvi, A.; Maheshwari, P.; Zhang, Z.; Santhanam, K.; Vardhamanan, S.; Haq, S.; Sharma, A.; Joshi, T. T.; Moazam, H.; et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Kim, T. S.; Lee, Y.; Shin, J.; Kim, Y.-H.; and Kim, J. 2024. Evallm: Interactive evaluation of large language model prompts on user-defined criteria. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–21.

Lybarger, K.; Dobbins, N. J.; Long, R.; Singh, A.; Wedgeworth, P.; Uzuner, Ö.; and Yetisgen, M. 2023. Leveraging natural language processing to augment structured social determinants of health data in the electronic health record. *Journal of the American Medical Informatics Association*, 30(8): 1389–1397.

Mahbub, M.; Dams, G. M.; Srinivasan, S.; Rizy, C.; Danciu, I.; Trafton, J.; and Knight, K. 2024. Leveraging large language models to extract information on substance use disorder severity from clinical notes: a zero-shot learning approach. *arXiv preprint arXiv:2403.12297*.

Majumdar, A.; Zhang, W.; Prawal, K.; and Yadav, A. 2025. The Hardness of Achieving Impact in AI for Social Impact

Research: A Ground-Level View of Challenges & Opportunities. *arXiv preprint arXiv:2506.14829*.

Philip, C.; Akshay, S.; Alex, J. G.; Yeasul, K.; Momsen Reincke, S.; Lichy, H.; Ben, D.; Sadeghi, M. A.; Abdel-Badih, A.; Marc, G.; David, S.; Andrew, A. L.; Caitlin, E. C.; Brad, B.; Mahir, A. S.; Hong, H. J.; Teresa, P. N.; Mohammad, R. R.; Komal, K.; Mark, A. B.; James, C. M.; Roya, S.; Stephen, P. M.; Dev, D.; James, X.; Ellen, Y. W.; Clifford, A. S.; Nigam, S.; and Nima, A. 2025. VeriFact: Verifying facts in LLM-generated clinical text with electronic health records. *arXiv [cs.AI]*.

Pryzant, R.; Iter, D.; Li, J.; Lee, Y. T.; Zhu, C.; and Zeng, M. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Reza, M.; Anastasopoulos, I.; Bhandari, S.; and Pardos, Z. A. 2025. PromptHive: Bringing subject matter experts back to the forefront with collaborative prompt engineering for educational content creation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–22.

Shah, C. 2023. From prompt engineering to prompt science with human in the loop. *arXiv [cs.HC]*.

Shankar, S.; Chopra, B.; Hasan, M.; Lee, S.; Hartmann, B.; Hellerstein, J.; Parameswaran, A.; and Wu, E. 2025. Steering semantic data processing with docwrangler. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, 1–18.

Shankar, S.; Zamfirescu-Pereira, J.; Hartmann, B.; Parameswaran, A.; and Arawjo, I. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 1–14.

Shin, T.; Razeghi, Y.; Logan IV, R. L.; Wallace, E.; and Singh, S. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Shneiderman, B. 2020. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6): 495–504.

Sivaraman, V.; Vaishampayan, A.; Li, X.; Buck, B. R.; Ma, Z.; Boyce, R. D.; and Perer, A. 2025. Tempo: Helping data scientists and domain experts collaboratively specify predictive modeling tasks. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–18. New York, NY, USA: ACM.

Subramonyam, H.; Pea, R.; Pondoc, C.; Agrawala, M.; and Seifert, C. 2024. Bridging the gulf of envisioning: Cognitive challenges in prompt based interactions with LLMs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, volume 31, 1–19. New York, NY, USA: ACM.

Szymanski, A.; Gebreegziabher, S. A.; Anuyah, O.; Metoyer, R. A.; and Li, T. J.-J. 2024. Comparing criteria development across domain experts, lay users, and models in large language model evaluation. *arXiv preprint arXiv:2410.02054*.

Wang, X.; Kim, H.; Rahman, S.; Mitra, K.; and Miao, Z. 2024. Human-LLM collaborative annotation through effective verification of LLM labels. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–21. New York, NY, USA: ACM.

Williams, C. Y.; Bains, J.; Tang, T.; Patel, K.; Lucas, A. N.; Chen, F.; Miao, B. Y.; Butte, A. J.; and Kornblith, A. E. 2025a. Evaluating large language models for drafting emergency department encounter summaries. *PLOS digital health*, 4(6): e0000899.

Williams, C. Y.; Subramanian, C. R.; Ali, S. S.; Apolinario, M.; Askin, E.; Barish, P.; Cheng, M.; Deardorff, W. J.; Donthi, N.; Ganeshan, S.; et al. 2025b. Physician-and large language model–generated hospital discharge summaries. *JAMA Internal Medicine*.

Yang, C.; Wang, X.; Lu, Y.; Liu, H.; Le, Q. V.; Zhou, D.; and Chen, X. 2023. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Yu, Z.; Peng, C.; Yang, X.; Dang, C.; Adekkanattu, P.; Gopal Patra, B.; Peng, Y.; Pathak, J.; Wilson, D. L.; Chang, C.-Y.; Lo-Ciganic, W.-H.; George, T. J.; Hogan, W. R.; Guo, Y.; Bian, J.; and Wu, Y. 2024. Identifying social determinants of health from clinical narratives: A study of performance, documentation ratio, and potential bias. *J. Biomed. Inform.*, 153: 104642.

Yuksekgonul, M.; Bianchi, F.; Boen, J.; Liu, S.; Huang, Z.; Guestrin, C.; and Zou, J. 2024. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*.

Zhang, T.; Ladhak, F.; Durmus, E.; Liang, P.; McKeown, K.; and Hashimoto, T. B. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12: 39–57.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36: 46595–46623.

Zhou, Y.; Muresanu, A. I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; and Ba, J. 2022. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*.

Zhu, L.; Wang, X.; and Wang, X. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.