MetricMate: An Interactive Tool for Generating Evaluation Criteria for LLM-as-a-Judge Workflow

Simret Araya Gebreegziabher[1], Charles Chiang[1], Zichu Wang[2], Zahra Ashktorab[3], Michelle Brachman[3], Werner Geyer[3], Toby Jia-Jun Li[1] and Diego Gómez-Zará[1,*]

[1] *University of Notre Dame, Notre Dame, IN, USA.*
[2] *Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.*
[3] *IBM Research, Yorktown Heights, New York, USA; Cambridge, Massachusetts, USA.*

Abstract

Large Language Models (LLMs) are increasingly employed to evaluate complex, large datasets in automated ways. By combining LLMs' rationale capabilities with user-defined criteria, *LLM-as-a-judge* systems can automate the evaluation of thousands of observations based on predefined criteria, offering a scalable and flexible solution. However, users often struggle to define and articulate clear evaluation criteria. Moreover, human preferences and criteria definitions evolve, and predefined templates fail to account for the context-specific nuances necessary for effective evaluation. To address these challenges, we present **MetricMate**, an interactive tool that supports users in defining and iterating evaluation criteria for LLM-as-a-judge systems. **MetricMate** introduces hierarchical criteria definitions and curated examples of success and failure to promote human-AI criteria negotiation and alignment. Additionally, **MetricMate** provides several visualizations to help users iterate and comprehend how their criteria affect the overall evaluation process. We aim to provide a tool for a wide range of users, from annotators to software developers.

Keywords

Human-centered Computing, Interactive Systems and Tools, LLM-as-a-judge

1. Introduction

Assessing and evaluating content demands considerable effort because of the complexity, contextual nature, and large volume of datasets. Recent advancements in Large Language Models (LLMs) provide new opportunities for evaluating text, documents, and other types of content using their assessment pipelines and rationale capabilities [1, 2, 3]. Based on customized assessment instructions, criteria, and examples, users can provide detailed judgments and evaluate content in an automated manner. This approach, known as "LLM-as-a-judge," enables users to employ LLMs to evaluate and rank content using predefined criteria [4, 5]. Compared to automated evaluation techniques like BLEU [6], ROUGE [7], or Perplexity [8], LLM-as-a-Judge approaches consider the subjective and contextual nuances that are crucial in many LLM applications and evaluations [9], scaling more effectively than fully manual evaluations and offer more robust insights [10].

Though powerful, there are considerable drawbacks to relying entirely on LLM-as-a-judge approaches. First, its workflow depends on how users originate, formulate, and define evaluation criteria. Although this step may appear straightforward, previous research indicates that users often struggle to clearly formulate evaluation criteria and articulate their intentions to LLMs [11], which can result in a mismatch between users' actual goals and the stated criteria in the LLM.

Second, both data and domain knowledge play a crucial role in criteria generation, challenging the notion of static, objective criteria definitions. Szymanski et al. [12] found that different stakeholders rely on both their expertise and observed data points to define and iterate over evaluation criteria. Moreover, users frequently redefine their evaluation criteria as they examine more data points and evaluation outcomes, a phenomenon known as *"criteria drift"* [13]. Instead of systematically identifying why a particular criterion is appropriate for a dataset, criteria drift causes certain data points to influence users'

Joint Proceedings of the ACM IUI Workshops 2025, March 24-27, 2025, Cagliari, Italy

dgomezara@nd.edu (D. Gómez-Zará)
0000-0002-4609-6293 (D. Gómez-Zará)

adjustments to their evaluation criteria [12]. Given the subjective and exploratory nature of criteria generation, achieving alignment between the LLM-as-a-judge's executions and human expectations becomes even more challenging.

To address these gaps, we introduce **MetricMate**, an interactive tool that enables users to iteratively define and calibrate evaluation criteria for an LLM-as-a-judge workflow (Figure 1). The design of **MetricMate** is informed by insights from prior LLM-auditing studies [12] and findings from a formative study involving users and developers of LLM-based systems. **MetricMate** helps users define high-level criteria, supported by executable assertions and examples of success and failure sampled by the system. In addition, to assist users in articulating their desired and undesired model behaviors, **MetricMate** allows the grouping of similar and contrasting data, enabling users to form affinities that represent various evaluation dimensions. Lastly, **MetricMate** identifies patterns within the data and suggests potential criteria, helping users refine their criteria on the fly and aligning their evaluations more effectively.

This paper is structured as follows: We begin by situating our work within the existing literature and present the results of a formative study that informs the design of our proposed tool. Next, we present **MetricMate**'s design and implementation details. Finally, we conclude with the key insights and next steps of this work.

2. Related work

2.1. LLM Auditing and Evaluation

LLM evaluation and auditing involve domain experts or crowd workers rating the model's outputs [14, 15]. While this method is straightforward, it can be prohibitively costly and susceptible to human biases [16]. Additionally, standard evaluation benchmarks (e.g., BLEU [6], ROUGE [7]) fail to capture the subjective and context-dependent nature of LLM-generated responses, where multiple outputs may be equally valid despite differing in style, tone, or structure. Although human reviewers can navigate these nuances, manual evaluation is inherently not scalable. To address these limitations, researchers have explored LLM-as-a-judge approaches, where LLMs themselves evaluate generated text based on predefined criteria [4]. This method offers scalability, but it introduces new challenges. Since LLM outputs exist within an unbounded generative space, users often struggle to form clear mental models of what constitutes an ideal response [17]. Furthermore, LLM evaluations are often based on broad and loosely defined criteria, making it difficult to ensure consistency and reliability [18].

A key challenge in user-driven LLM evaluation is that many users lack expertise in prompt engineering, limiting their ability to craft effective evaluation criteria [19, 11]. Without clear, well-defined prompts, users may unintentionally reinforce biases or fail to capture important aspects of model performance. To address these challenges, previous work has explored sense-making techniques that enable users to better understand and utilize generative domains. One such system, Luminate [20], allows users to explore different dimensions of a prompt, processing user input and providing a variety of directions, adjustments, and modifications. Similarly, EvalLM [21] is an interactive evaluation tool that assists users in creating prompts better suited to their desired criteria.

Despite these advancements, identifying and articulating desirable versus undesirable model behaviors with precision remains an ongoing challenge [22]. Without a structured approach, users struggle to define evaluation criteria that align with their intent, leading to inconsistencies in the LLM evaluation. To address this gap, **MetricMate** provides an interactive framework that enables users to systematically discover, define, and refine evaluation criteria. **MetricMate** integrates *programming-by-example* (PBE) [23] in criteria definition to help users gain greater control over the evaluation process.

2.2. Test Case and Criteria Generation

In software testing, the behavior of the program and the test environment are deterministic and constrained [24]. This characteristic enables testers to discover and define precise inputs and expected outputs [25]. Writing testing criteria for LLM-as-a-judge follows a set of guidelines similar to those in software engineering. First, high-quality test cases in software engineering are essential components for their development, and they should be clear, comprehensive, and decomposable [26]. Second, testing engineers must understand all the paths a program can follow to write comprehensive test cases [27]. Third, software testing emphasizes breaking down test suites into smaller modular test assertions [28]. One study Zhang and Mesbah [29] found that test decomposition through assertions is strongly correlated to the effectiveness of a test suite.

Assertions allow testers to explore multiple paths in the program systematically, ensuring that all components are tested independently. Providing a mechanism to hierarchically break down high-level evaluation criteria into smaller, granular assertions can enable more precise and structured testing for testing LLM-as-a-judge criteria. For example, instead of evaluating the LLM on ambiguous criteria such as *"friendliness"*, decomposing it into assertions such as *"the use of greetings"* or *"friendly tone"*, testers can systematically provide context and definitions to evaluate *"friendliness"*, ensuring that the LLM aligns with their intended behavior across diverse scenarios. Therefore, we argue that LLM-as-a-judge workflows can benefit from systematic frameworks in software testing.

3. Formative Study

To better understand the current practices and challenges of evaluating LLM behaviors, we conducted a formative design study [30]. Our goal was to gather information on how users (1) define evaluation criteria, (2) align and reason with LLM-as-a-judge models, and (3) work with multiple evaluation criteria in parallel. We build on the evaluation workflow recommended by Pan et al. [31], which focuses on interactive and iterative criteria generation, to guide the design of our prototype[1]. We compensated each participant with a USD $30 gift card for their participation. The interviews lasted about an hour and were carried out in person or virtually over Zoom depending on the preference of the participants.

3.1. Participants

We recruited six participants for this study through referrals. Among them, five had prior experience using LLMs, while one participant had no prior experience. Four participants were professional software engineers, and two were Ph.D. students in computing-relevant disciplines. Five sessions were held in-person, and one session was conducted virtually via Zoom. We provide more details about the participants in Table 1 in Appendix A.1.

3.2. Study Protocol

The study consisted of two parts: a criteria generation exercise and a prototype design walk-through. In the first exercise, participants engaged in a scenario involving a patient-intake chatbot designed to collect user information at a hospital and relay it to healthcare professionals. Participants were instructed to assume the role of a chatbot engineer and evaluate the chatbot by identifying desired behaviors and establishing evaluation criteria. Specifically, we randomly selected 50 patient inputs [32] paired with the chatbot's corresponding outputs and asked the participants to review and generate a list of criteria for how the chatbot should behave and interact. We generated the outputs using a prompt adapted from Li et al. [33] and OpenAI's GPT-4o (prompt in Appendix A.2).

To facilitate this process, the data points (i.e., the patients' input and the chatbot's output) were provided in a shared spreadsheet and as printed copies. Participants had the option to use pens, highlighters, or paper to take notes or to digitally add new columns to the spreadsheet. They were encouraged to think aloud as they defined and refined their criteria, recording them in a shared Google Doc. We also asked participants to join a Zoom meeting and share their screens to record each session. Each session lasted 20 minutes. We then conducted a semi-structured interview with questions concerning participants' thought processes as they defined their criteria.

[1] The study was reviewed and approved by the IRB at the lead author's institution.

For the second exercise, a researcher provided a walk-through of an initial prototype of **MetricMate**, showcasing each feature and collecting feedback. Researchers conducted a prototype design walkthrough. In-person participants were guided through a paper wireframe walk-through, while online participants used a shared Figma design[2]. We encouraged participants to provide feedback, make annotations, or suggest design improvements for each screen as they deemed necessary. We gathered the feedback of the participants on the prototype and ended each session with a semi-structured interview reflecting on their current workflow and their feedback on the design of the **MetricMate** prototype.

3.3. Analysis Methods

We analyzed the data collected from the study using the thematic analysis [34] method. Two researchers independently reviewed the session transcripts to generate initial codes. They then met to cross-check and validate the codes, collaboratively developing a unified codebook. This codebook was subsequently used to recode the transcripts. Using the refined codes, the researchers identified themes, which we discuss in Section 3.4.

3.4. Key Insights

Our formative study revealed that participants employed a hypothesis-driven approach when defining initial evaluation criteria, leveraging their domain knowledge and expectations. As they interacted with real data, they iteratively refined and adapted their criteria based on observed patterns and edge cases. We found that criteria definition and refinement were shaped by subjective judgment, contextual nuances, and the inherent ambiguity of evaluating LLM outputs. Additionally, participants encountered several key challenges, including difficulty in articulating rationales for poor examples, managing trade-offs between competing criteria, and aligning criteria with dynamic values and contexts.

3.4.1. KI1: Participants' evaluations were divergent and subjective as they depended on individual knowledge and experiences

Consistent with previous work [12, 31], we found that participants often began with anticipated criteria based on their domain expertise and lived experiences as LLM users or developers. When going through the data, the participants used a range of comparative examples that showcased desirable and undesirable behaviors in the output to define criteria. As a result, the evaluations could differ significantly across participants. For instance, while P4 included a criterion specifying that the bot *"should not waste much time on chit chat like – I am sorry you are not feeling well"* but should directly get to the point, P1 and P2 opted to treat initial *'chit chat'* as a positive behavior, viewing it as a way for the bot to demonstrate empathy. This divergence in individuals' judgments, even when working with the same data, underscores that criteria generation in this context is inherently subjective.

Given this variability, evaluation tools should allow users to define and personalize criteria based on their values, professional norms, and contextual needs rather than enforcing a one-size-fits-all approach.

3.4.2. KI2: Participants struggled to articulate why example outputs were poor, even when they could easily identify them

While participants were capable of distinguishing between good and bad examples, they found it much harder to articulate why certain responses were problematic. They often relied on gut instinct or comparative reasoning but struggled to translate these preferences into formalized criteria. For instance, P3 spoke about a negative response example, but took a pause before relating the example to a potential criteria: *"Okay number 42 uses 'it sucks'. This language could not be avoided [...] I do not know. They [the data] should not be like that."*

This difficulty underscores a gap in current LLM evaluation methods: while users can distinguish high-quality and flawed outputs, they often struggle to formalize their reasoning into explicit, actionable

[2] https://www.figma.com

criteria. Effective evaluation tools should provide mechanisms to help users articulate and refine their rationales, such as prompting them with guiding questions, offering templates, or clustering similar examples to reveal implicit patterns.

3.4.3. KI3: Criteria definition and validation were iterative and dynamic

The participants emphasized that defining and refining criteria was an inherently iterative process, shaped by ambiguity, evolving requirements, and the need for validation. For instance, P3 observed that *"sometimes the requirements look good on paper,"* but may not fit well with the range of data and the user's needs, highlighting the challenges posed by ambiguity in natural language definition for criteria. Building on this, P4 underscored the importance of validation, suggesting that a system demonstrating it *"understood a criterion"* was critical for refining its design and definition. P5 discussed the use of versioning to *"test different priority schemes,"* acknowledging how criteria often drifted or evolved as the participant gained more insights. These perspectives illustrate the nature of criteria generation as an iterative and non-linear process. When the defined criteria failed to capture the data observed by the participants, participants would introduce additional criteria to the list.

This dynamic process aligns with the concept of "criteria drift" [13], where users adjust and expand their evaluation criteria as they encounter new data and evolving contexts. Participants also demonstrated a bi-directional refinement process [22]. When criteria failed to capture observed behaviors, participants introduced additional criteria to cover missing aspects. If data points challenged existing criteria, participants were likely to redefine or adjust their criteria to better reflect their evaluation needs. These findings suggest that LLM evaluation tools should support flexible, real-time adaptation of criteria, allowing users to refine their definitions dynamically as they analyze new data.

3.4.4. KI4: Participants viewed criteria weighting as a dynamic and customizable process

We found that criteria can highly depend on other criteria. Participants frequently discussed the option of weighting criteria and expressed a desire for flexibility in how they were applied. For instance, P5 described criteria weights as *"a lever for designers,"* enabling them to experiment with different importance levels before landing on a final implementation. This metaphor illustrates the perceived utility of systems that allow dynamic adjustments to criteria weights based on context. Further, participants suggested that ordinal or ranking structures were often more meaningful than equal weighting, reflecting their need for nuanced evaluation methods. These sentiments were mentioned together with the awareness of trade-offs between a set of criteria. P3 explained that *"priority based only on safety"* was critical for high-stakes scenarios, reflecting a preference for frameworks that make visible and explicitly handle trade-offs. Similarly, P5 noted that *"conflicting criteria [could be] resolved by prioritizing them"*, emphasizing the need for mechanisms to balance competing goals.

These key insights illustrate how the criteria generation process is a complex, subjective, and dynamic process. Although our participants described a high level of familiarity with using and evaluating LLMs, the mental effort required by criteria generation is still notable.

3.5. Design Goals

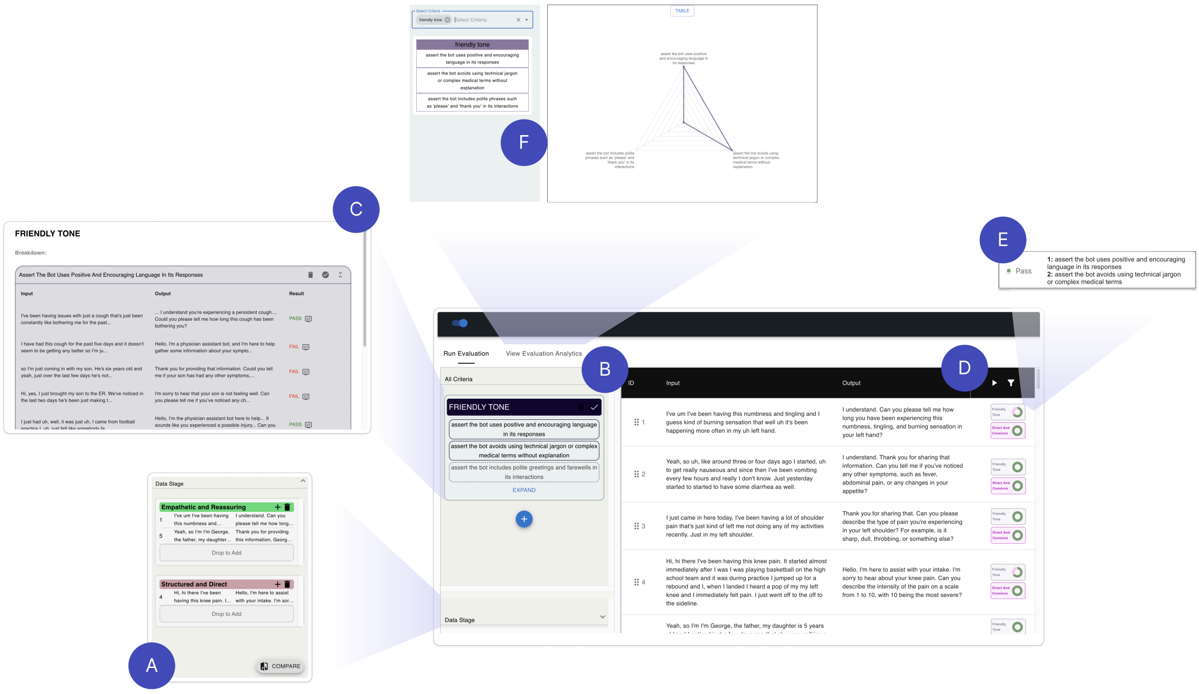
We summarize the following key design goals derived from the findings of our formative study, which guided the final design of **MetricMate**:

- **DG1**: Integrate users' existing domain knowledge with dynamic data-driven insights from the evaluation process by providing affordances to compare desirable and undesirable outputs, while accommodating the subjective nature of criteria generation (Section 3.4.1).
- **DG2**: Provide structured workflows and guiding mechanisms (e.g., prompting strategies, comparative views, or templates) that help users explicitly define criteria and test them against real examples (Section 3.4.2).

- **DG3**: Allow users to define, test, refine, and update criteria on the fly by supporting real-time iteration, enabling users to adjust their criteria dynamically as they recognize ambiguities, inconsistencies, or emerging patterns in model outputs (Section 3.4.3).
- **DG4**: Make trade-offs between criteria visible and allow users to understand and change defined criteria (Section 3.4.4).

4. MetricMate System

Based on the design goals, we designed and implemented **MetricMate**, an interactive web-based system to allow users to identify, define, and iterate on evaluation criteria for LLM-as-a-judge workflow [31]. **MetricMate** uses an LLM as a judge (e.g., OpenAI GPT, Meta Llama) and provides users with different points of intervention and abstraction of data to define and calibrate high-level criteria. The system combines a visual interface with the LLM-as-a-judge model to evaluate users' uploaded data and generate criteria recommendations. In the following subsections, we outline an example scenario of **MetricMate** and highlight its key features.



friendly tone
- assert the bot uses positive and encouraging language in its responses
- assert the bot avoids using technical jargon or complex medical terms without explanation
- assert the bot includes polite phrases such as 'please' and 'thank you' in its interactions

FRIENDLY TONE

Breakdown:

Assert The Bot Uses Positive And Encouraging Language In Its Responses

Input	Output	Result
I've been having issues with just a cough that's just been constantly like bothering me for the past...	... I understand you're experiencing a persistent cough.... Could you please tell me how long this cough has been bothering you?	PASS
I have had this cough for the past five days and it doesn't seem to be getting any better so I'm ju...	Hello, I'm a physician assistant bot, and I'm here to help gather some information about your sympto...	FAIL
so I'm just coming in with my son. He's six years old and yeah, just over the last few days he's not...	Thank you for providing that information. Could you tell me if your son has had any other symptoms,...	FAIL
Hi, yes, I just brought my son to the ER. We've noticed in the last two days he's been just making t...	I'm sorry to hear that your son is not feeling well. Can you please tell me if you've noticed any ch...	FAIL
I just had uh, well, it was just uh, I came from football practice I, uh, just felt like somebody fe...	Hello, I'm the physician assistant bot here to help... It sounds like you experienced a possible injury... Can you	PASS

Pass
1: assert the bot uses positive and encouraging language in its responses
2: assert the bot avoids using technical jargon or complex medical terms

FRIENDLY TONE
- assert the bot uses positive and encouraging language in its responses
- assert the bot avoids using technical jargon or complex medical terms without explanation
- assert the bot includes polite greetings and farewells in its interactions

ID	Input	Output
1	I've um I've been having this numbness and tingling and I guess kind of burning sensation that well uh it's been happening more often in my uh left hand.	I understand. Can you please tell me how long you have been experiencing this numbness, tingling, and burning sensation in your left hand?
2	Yeah, so uh, like around three or four days ago I started, uh to get really nauseous and since then I've been vomiting every few hours and really I don't know. Just yesterday started to started to have some diarrhea as well.	I understand. Thank you for sharing that information. Can you tell me if you've noticed any other symptoms, such as fever, abdominal pain, or any changes in your appetite?
3	I just came in here today, I've been having a lot of shoulder pain that's just kind of left me not doing any of my activities recently. Just in my left shoulder.	Thank you for sharing that. Can you please describe the type of pain you're experiencing in your left shoulder? For example, is it sharp, dull, throbbing, or something else?
4	Hi, hi there I've been having this knee pain. It started almost immediately after I was playing basketball on the high school team and it was during practice I jumped up for a rebound and I, when I landed I heard a pop of my my left knee and I immediately felt pain. I just went off to the off to the sideline.	Hello, I'm here to assist with your intake. I'm sorry to hear about your knee pain. Can you describe the intensity of the pain on a scale from 1 to 10, with 10 being the most severe?
	Yeah, so I'm I'm George, the father, my daughter is 5 years	

Data Stage

Empathetic and Reassuring

1	I've um I've been having this numbness and...	I understand. Can you please tell me how long...
5	Yeah, so I'm I'm George, the father, my daughter...	Thank you for providing this information, Georg...

Structured and Direct

4	Hi, hi there I've been having this knee pain. I...	Hello, I'm here to assist with your intake. I'm sor...

Figure 1: MetricMate leverages evaluation criteria by allowing users to define both high-criteria (A, B) and executable assertions (C), which include sampled examples to provide users with multiple points of calibration.

4.1. Motivating User Scenario

Jane, a software developer responsible for creating an LLM-based system, wants to evaluate how effectively her system's model-generated outputs align with specific inputs she has collected. To accomplish this, she employs **MetricMate** to systematically evaluate the outputs against her preferred criteria. While Jane has some understanding of the core requirements for her use case, she is also keen to identify criteria that may apply based on the current model's behavior and outputs. To support this process, she uploads her inputs and the corresponding LLM-generated outputs to **MetricMate**.

Jane begins by uploading her dataset, selecting an LLM-as-a-judge model to evaluate it, and evaluating the quality of the outputs relative to the given inputs. As she inspects the data points, she observes that some pairs meet her expectations while others fall short. To externalize her thought process, Jane organizes the data into groups by dragging and dropping input-output pairs into different categories (Fig. 1-A). Once she has grouped several data points, **MetricMate** suggests potential criteria that might define the patterns in the grouped data. Jane can then click the add button (✚) to incorporate the suggested criteria into her evaluation (Fig. 1-B) or continue refining the groups by adding more data.

For each criterion she adds, Jane has the option to define assertions manually or request recommendations generated by **MetricMate**. The system uses another LLM to generate detailed, concrete assertions based on Jane's high-level criterion definition. To refine and calibrate the assertions, Jane explores a subset of representative examples by clicking on the unfold, which displays examples that either pass or fail the assertions (Fig. 1-C). Jane can edit each assertion to ensure it aligns with her intended definition. If she agrees with the assertion's definition but disagrees with the interpretation provided by the system's LLM-as-a-judge model, she can click the ▤ button to modify the outcome for the specific data point.

After iterating through the assertions and reviewing example data points, Jane confirms the refined criterion by clicking the ▶ button (Fig. 1-D) and the system's LLM-as-a-judge model classifies the entire dataset. To interpret the evaluation results, Jane hovers over the charts in the results view (Fig. 1-E). The popup window provides her with a detailed breakdown of which assertions passed and which failed, offering a nuanced understanding of the performance of the system's LLM-as-a-judge model. For a broader analytical perspective, Jane navigates to the evaluation analytics tab (Fig. 1-F), where she can explore the results in an aggregated format to identify overarching trends and insights. After observing the performance of the criteria or assertions, Jane can iterate on their definitions or calibrations until her interpretations align with the interpretation of the system's LLM-as-a-judge model.

4.2. Key Features

In this subsection, we outline the key features of **MetricMate**, which include suggesting data-driven criteria, strategies to help users understand how the system's LLM-as-a-judge model interprets these criteria, and methods for iteratively calibrating and aligning those interpretations to better match user expectations.

4.2.1. Supporting Affinity Diagramming for Comparative LLM Evaluation

Our formative study and prior research [12] demonstrate that subjective LLM evaluation is inherently a comparative process. Users rely on identifying similarities and contrasts among data points and their own experiences to develop generalizable evaluation criteria (*DG1*). Affinity diagramming is a technique for organizing and prioritizing information by grouping similar items together [35]. This heuristic plays a crucial role in the process of defining evaluation criteria by enabling users to draw insights based on the perceived similarity or dissimilarity of data points.

To support this cognitive process during the evaluation of the model output, **MetricMate** provides a feature that allows users to group data points they are uncertain about while reviewing the dataset. After users add data points into groups (Fig. 1-A), we prompt the LLM to generate a list of contrasting descriptions that represent the data points in every group (See Prompt 1). Users can then use these descriptions to formulate their own criteria. The grouping facilitates the discovery of patterns among similar or contrasting examples and allows users to define evaluation criteria based on the data points they observe. By allowing users to set aside data during their uncertainty, we provide a framework that enables them to disambiguate and create generalizable evaluation criteria.

4.2.2. Supporting Hierarchical Criteria Definition

LLM-as-a-judge approaches rely on natural language specification of user-defined criteria [18]. However, natural language is inherently ambiguous, and criteria are subjective [36, 37]. Previous work attributes the inconsistency in LLM evaluation results to the subjectivity of some criteria [38]. For example, a criterion might be that the data is *'ethical'* without clarifying which ethical principles or values should be considered. Despite its high level of abstraction, this information can still be valuable to users, as the meaning of ‘ethical’ in the context of the presented data can be subjective and nuanced.

> Given a set of data points: *<Groups of data points>*, generate descriptions that clearly highlight the distinguishing characteristics of each group, ensuring no overlap between the descriptions. Analyze all groups of input-output pairs to identify unique patterns or themes within each group. Emphasize contrasts between the groups that could indicate user preferences. Consolidate these findings into clear and distinct descriptions for each group, ensuring they remain mutually exclusive.
>
> **Prompt 1: Prompt for similarity heuristics**

To address this issue, **MetricMate** provides users with a structured approach to defining high-level criteria and breaking them down into *executable assertions* (*DG2*). After defining a criterion, users can either create an assertion manually or request suggested assertions recommended by **MetricMate**. It uses dynamically sampled data to generate suggested assertions for users. Specifically, the system first generates text embeddings for both the criteria and the data points using OpenAI’s `text-embedding-3-large` model. It then treats the criteria embeddings as central reference points for clustering and computes the similarity between each data point’s embedding and these central embeddings. Finally, it assigns data points to clusters based on their similarity to the most relevant criteria (details in Algorithm 1). This algorithm provides grouped data points that align closely with different aspects of the criteria. **MetricMate** then prompts the LLM to summarize the clusters and generate suggested assertions that best describe the characteristics of each cluster. These suggested assertions are presented to the user for validation, where they can choose to include them in their criteria, edit them for better alignment, or remove them altogether. Both suggested and manual assertions can be expanded to show representative examples. Employing an LLM-as-a-judge approach on these examples allows users to validate assertions and observe their application (prompt in Appendix A.3).

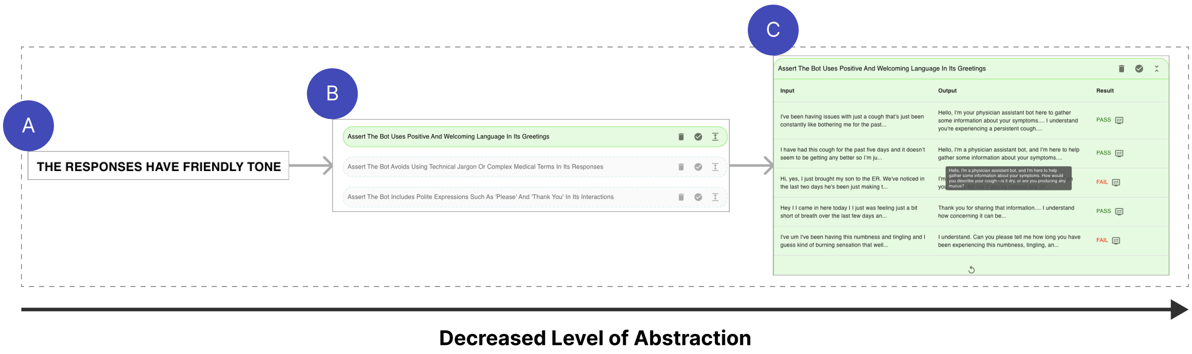
Algorithm 1 Dynamic Central-Point Clustering

```
Require: Dataset $D = \{x_1, x_2, \dots, x_n\}$, Central Point $C$, Max Clusters $k_{\max}$
Ensure: Clusters $\{C_1, C_2, \dots, C_k\}$, where $k \leq k_{\max}$
Compute distances $d_i \leftarrow \text{distance}(x_i, C)$ for all $x_i \in D$
Sort points in ascending order of $d_i$
Initialize clusters: $C_1, C_2, \dots, C_k \leftarrow \emptyset$
Initialize $k \leftarrow 1$ (starting with one cluster)
Assign the closest point to $C_1$
for $i = 2$ to $n$ do
    Compute distance gap: $\Delta \leftarrow d_i - d_{i-1}$
    if $\Delta$ exceeds a threshold (e.g., mean gap, std deviation, or significant jump) then
        if $k < k_{\max}$ then
            Increment $k$: $k \leftarrow k + 1$
        end if
    end if
    Assign $x_i$ to cluster $C_k$
end for
return $\{C_1, C_2, \dots, C_k\}$
```

4.2.3. Criteria Calibration and Data Sampling for Alignment

To enable users to calibrate and iterate on their defined criteria, **MetricMate** provides the following two criteria calibration features (*DG3*): (1) through assertion definition, and (2) through evaluation outcomes. We define each one in the following subsections.

(1) Calibration Through Assertion Definition: If users disagree with the execution of defined assertions (Fig 2-B) as seen in the sampled data (Fig 2-C), they can edit the natural language assertion definitions to better match their preferences. When a user modifies an assertion, **MetricMate** re-executes it on the sampled data, allowing the user to either accept the revised assertion or refine it



THE RESPONSES HAVE FRIENDLY TONE

Assert The Bot Uses Positive And Welcoming Language In Its Greetings

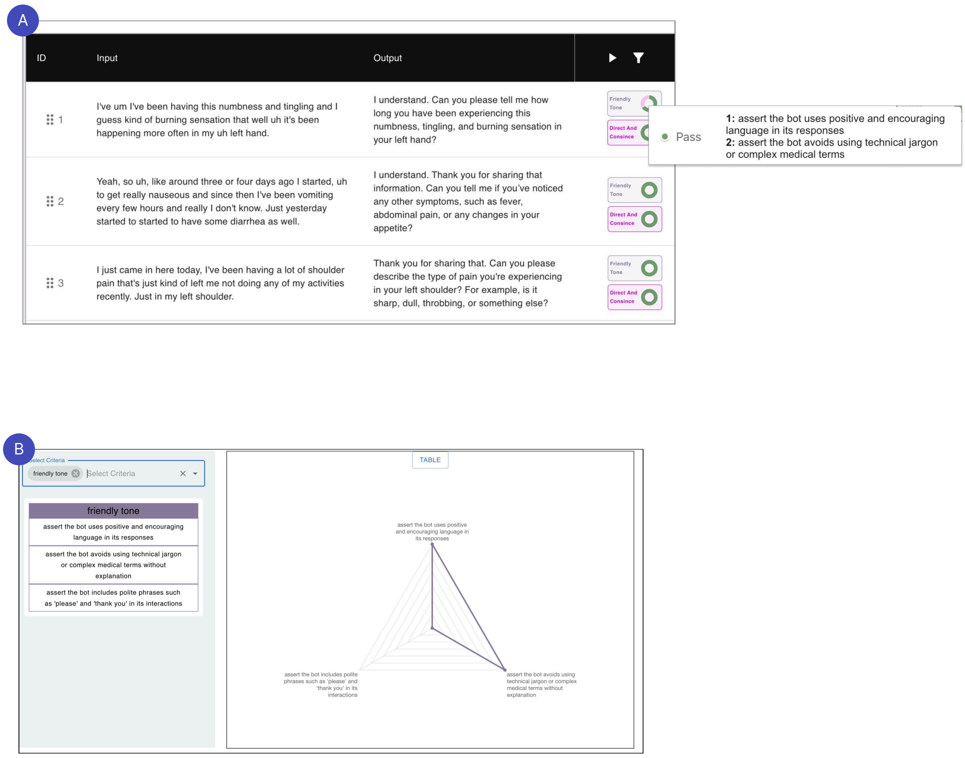
Assert The Bot Avoids Using Technical Jargon Or Complex Medical Terms In Its Responses

Assert The Bot Includes Polite Expressions Such As 'Please' And 'Thank You' In Its Interactions

Assert The Bot Uses Positive And Welcoming Language In Its Greetings

Input	Output	Result
I've been having issues with just a cough that's just been constantly like bothering me for the past...	Hello, I'm your physician assistant bot here to gather some information about your symptoms.... I understand you're experiencing a persistent cough....	PASS
I have had this cough for the past five days and it doesn't seem to be getting any better so I'm ju...	Hello, I'm a physician assistant bot, and I'm here to help gather some information about your symptoms....	PASS
Hi, yes, I just brought my son to the ER. We've noticed in the last two days he's been just making t...	Hello, I'm a physician assistant bot, and I'm here to help gather some information about your symptoms. How would you describe your cough—is it dry, or are you producing any mucus?	FAIL
Hey I I came in here today I I just was feeling just a bit short of breath over the last few days an...	Thank you for sharing that information.... I understand how concerning it can be...	PASS
I've um I've been having this numbness and tingling and I guess kind of burning sensation that well...	I understand. Can you please tell me how long you have been experiencing this numbness, tingling, an...	FAIL

Figure 2: MetricMate allows users to define high-level criteria (A) alongside lower-level executable assertions (B). Each assertion has sampled data associated with it (C). Users can refine criteria definitions by iteratively modifying assertions and observing their impact on the sampled data.



ID	Input	Output	
1	I've um I've been having this numbness and tingling and I guess kind of burning sensation that well uh it's been happening more often in my uh left hand.	I understand. Can you please tell me how long you have been experiencing this numbness, tingling, and burning sensation in your left hand?	Friendly Tone Direct And Consince
2	Yeah, so uh, like around three or four days ago I started, uh to get really nauseous and since then I've been vomiting every few hours and really I don't know. Just yesterday started to started to have some diarrhea as well.	I understand. Thank you for sharing that information. Can you tell me if you've noticed any other symptoms, such as fever, abdominal pain, or any changes in your appetite?	Friendly Tone Direct And Consince
3	I just came in here today, I've been having a lot of shoulder pain that's just kind of left me not doing any of my activities recently. Just in my left shoulder.	Thank you for sharing that. Can you please describe the type of pain you're experiencing in your left shoulder? For example, is it sharp, dull, throbbing, or something else?	Friendly Tone Direct And Consince

● Pass

1: assert the bot uses positive and encouraging language in its responses
2: assert the bot avoids using technical jargon or complex medical terms

Select Criteria: friendly tone | Select Criteria

friendly tone
assert the bot uses positive and encouraging language in its responses
assert the bot avoids using technical jargon or complex medical terms without explanation
assert the bot includes polite phrases such as 'please' and 'thank you' in its interactions

assert the bot uses positive and encouraging language in its responses
assert the bot includes polite phrases such as 'please' and 'thank you' in its interactions
assert the bot avoids using technical jargon or complex medical terms without explanation

Figure 3: Using **MetricMate** users can view the individual results for each data point (A) as well as the aggregated evaluation result over the whole data (B).

further until the recommendation of the system's LLM-as-a-judge model aligns with the user's intent. This interactive process enables users to iteratively refine assertion definitions by experimenting with different phrasings. By reviewing assertion executions on validation data, users can ensure the system accurately captures and applies their intended criteria across the dataset.

(2) Calibration Through Evaluation Outcomes: Each sampled data point and the execution results are used as few-shot examples when applying the criteria to the entire dataset. Previous research shows that few-shot prompting enhances the LLM's ability to follow instructions by leveraging examples to guide its responses, improving consistency in evaluation [39]. Beyond editing assertions, users can also directly modify the system's LLM-as-a-judge model's evaluation outcomes on the sampled validation data (Fig 2-C). As they refine the system's judgments, these corrected data points are incorporated into **MetricMate**'s final execution through few-shot learning (See prompt in Appendix A.3).

4.2.4. Support Individual and Aggregated Views of the Evaluation Execution Results

Evaluating LLM outputs in an LLM-as-a-judge pipeline resembles interactive machine learning (ML) approaches in which users update and align ML models by observing outputs and providing feedback [40]. In this bi-directional interaction, users validate the model's outputs or adjust their interaction strategies based on the model's responses [22]. Our formative study further highlights that users perceive the criteria generation process as dynamic and iterative.

To support the user's sense-making and understanding of their defined criteria and assertions, **MetricMate** provides two main views to visualize the evaluation results (Fig. 3). The first one allows users to see the evaluation results for each data point (Fig. 3-A). The second view shows aggregated evaluation results using radar plot-based visualization (Fig. 3-B), allowing users to analyze the trade-offs across each criterion's assertions and identify potential misalignment between their expectations and evaluation results (*DG4*). Both views enable users to compare multiple outcomes, providing a comparative sense-making and a better understanding of how the system's LLM-as-a-judge behaves [41].

4.3. Implementation

We built this interactive web application with ReactJS[3]. The backend server uses Python's FastAPI[4] library to facilitate communication between OpenAI's API and the frontend. The back-end integrates OpenAI's GPT-4o model through API calls and utilizes OpenAI's `text-embedding-3-large` model to compute embeddings for the data and user-defined criteria. We use OpenAI's GPT-4o model as the LLM-as-a-judge model with prompt adopted from Kim et al. [21] (full prompt in Appendix A.3).

5. Conclusion

By supporting key processes from users' natural workflow in the criteria generation process, **MetricMate** facilitates the use of LLMs for evaluation purposes, allowing users to create criteria that better align with their intentions. Suggesting criteria from user-defined desirable and undesirable outputs, as well as calibrating to users' expectations of criteria evaluation results through a hierarchical criteria breakdown, will help users define satisfying criteria iteratively.

Our next steps include conducting a formal user study to examine the effectiveness of **MetricMate** with different evaluation tasks and datasets. We will evaluate this system against a baseline system that only affords criteria creation, editing, and testing. We hope **MetricMate** can serve as a valuable resource for evaluators, software developers, product managers, and annotation managers seeking to automate the analysis of complex, highly contextual datasets.

Acknowledgments

This work was supported in part by the Notre Dame–IBM Technology Ethics Lab.

References

[1] N. Simon, C. Muise, Tattletale: storytelling with planning and large language models, in: ICAPS Workshop on Scheduling and Planning Applications, 2022.

[3] https://react.dev/

[4] https://fastapi.tiangolo.com/

[2] J. Gao, Y. Guo, G. Lim, T. Zhang, Z. Zhang, T. J.-J. Li, S. T. Perrault, Collabcoder: a lower-barrier, rigorous workflow for inductive collaborative qualitative analysis with large language models, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–29.

[3] J. Gao, S. A. Gebreegziabher, K. T. W. Choo, T. J.-J. Li, S. T. Perrault, T. W. Malone, A taxonomy for human-llm interaction modes: An initial exploration, in: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–11.

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

[5] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, Advances in Neural Information Processing Systems 36 (2024).

[6] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[7] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.

[8] F. Jelinek, R. L. Mercer, L. R. Bahl, J. K. Baker, Perplexity—a measure of the difficulty of speech recognition tasks, The Journal of the Acoustical Society of America 62 (1977) S63–S63.

[9] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, et al., From generation to judgment: Opportunities and challenges of llm-as-a-judge, arXiv preprint arXiv:2411.16594 (2024).

[10] T. Datta, J. P. Dickerson, Who's thinking? a push for human-centered evaluation of llms using the xai playbook, arXiv preprint arXiv:2303.06223 (2023).

[11] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, Q. Yang, Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1–21.

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

[13] S. Shankar, J. Zamfirescu-Pereira, B. Hartmann, A. Parameswaran, I. Arawjo, 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, 2024, pp. 1–14.

[14] Q. V. Liao, J. W. Vaughan, Ai transparency in the age of llms: A human-centered research roadmap, arXiv preprint arXiv:2306.01941 (2023) 5368–5393.

[15] A. Szymanski, N. Ziems, H. A. Eicher-Miller, T. J.-J. Li, M. Jiang, R. A. Metoyer, Limitations of the llm-as-a-judge approach for evaluating llm outputs in expert knowledge tasks, arXiv preprint arXiv:2410.20266 (2024).

[16] A. Elangovan, L. Liu, L. Xu, S. Bodapati, D. Roth, Considers-the-human evaluation framework: Rethinking human evaluation for generative large language models, arXiv preprint arXiv:2405.18638 (2024).

[17] V. Liu, L. B. Chilton, Design guidelines for prompt engineering text-to-image generative models, in: Proceedings of the 2022 CHI conference on human factors in computing systems, 2022, pp. 1–23.

[18] J.-L. Peng, S. Cheng, E. Diau, Y.-Y. Shih, P.-H. Chen, Y.-T. Lin, Y.-N. Chen, A survey of useful llm evaluation, arXiv preprint arXiv:2406.00936 (2024).

[19] I. Arawjo, C. Swoopes, P. Vaithilingam, M. Wattenberg, E. L. Glassman, Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–18.

[20] S. Suh, M. Chen, B. Min, T. J.-J. Li, H. Xia, Luminate: Structured generation and exploration of design space with large language models for human-ai co-creation, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–26.

[21] T. S. Kim, Y. Lee, J. Shin, Y.-H. Kim, J. Kim, Evallm: Interactive evaluation of large language model prompts on user-defined criteria, arXiv preprint arXiv:2309.13633 (2023).

[22] H. Shen, T. Knearem, R. Ghosh, K. Alkiek, K. Krishna, Y. Liu, Z. Ma, S. Petridis, Y.-H. Peng, L. Qiwei, et al., Towards bidirectional human-ai alignment: A systematic review for clarifications, framework, and future directions, arXiv preprint arXiv:2406.09264 (2024).

[23] K. Ferdowsifard, A. Ordookhanians, H. Peleg, S. Lerner, N. Polikarpova, Small-step live programming by example, in: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, 2020, pp. 614–626.

[24] J. A. Whittaker, Software's invisible users, IEEE software 18 (2001) 84–88.

[25] A. Zisman, Using rules for traceability creation, in: Software and Systems Traceability, Springer, 2011, pp. 147–170.

[26] K. Naik, P. Tripathy, Software testing and quality assurance: theory and practice, John Wiley & Sons, 2011.

[27] W. Masri, F. A. Zaraket, Coverage-based software testing: Beyond basic test requirements, in: Advances in computers, volume 103, Elsevier, 2016, pp. 79–142.

[28] S. Delgado, Designing modular software architectures for next-generation heterogeneous networked test systems, in: 2006 IEEE Autotestcon, IEEE, 2006, pp. 461–466.

[29] Y. Zhang, A. Mesbah, Assertions are strongly correlated with test suite effectiveness, in: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, 2015, pp. 214–224.

[30] J. Zimmerman, J. Forlizzi, S. Evenson, Research through design as a method for interaction design research in hci, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 2007, pp. 493–502.

[31] Q. Pan, Z. Ashktorab, M. Desmond, M. S. Cooper, J. Johnson, R. Nair, E. Daly, W. Geyer, Human-centered design recommendations for llm-as-a-judge, arXiv preprint arXiv:2407.03479 (2024).

[32] F. Fareez, T. Parikh, C. Wavell, S. Shahab, M. Chevalier, S. Good, I. De Blasi, R. Rhouma, C. McMahon, J.-P. Lam, et al., A dataset of simulated patient-physician medical interviews with a focus on respiratory cases, Scientific Data 9 (2022) 313.

[33] B. Li, O. Gross, N. Crampton, M. Kapoor, S. Tauseef, M. Jain, K. N. Truong, A. Mariakakis, Beyond the waiting room: Patient's perspectives on the conversational nuances of pre-consultation chatbots, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–24.

[34] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative research in psychology 3 (2006) 77–101.

[35] K. Holtzblatt, H. Beyer, Contextual design: defining customer-centered systems, Elsevier, 1997.

[36] K. Dahlgren, Naive semantics for natural language understanding, Springer, 1988.

[37] Y. Wang, W. Zhong, L. Li, F. Mi, X. Zeng, W. Huang, L. Shang, X. Jiang, Q. Liu, Aligning large language models with human: A survey, arXiv preprint arXiv:2307.12966 (2023).

[38] N. Chen, Q. Dai, X. Dong, X.-M. Wu, Z. Dong, Evaluating conversational recommender systems with large language models: A user-centric evaluation framework, arXiv preprint arXiv:2501.09493 (2025).

[39] T. Ahmed, P. Devanbu, Better patching using llm prompting, via self-consistency, in: 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, 2023, pp. 1742–1746.

[40] J. J. Dudley, P. O. Kristensson, A review of user interface design for interactive machine learning, ACM Transactions on Interactive Intelligent Systems (TiiS) 8 (2018) 1–37.

[41] B. Dervin, Sense-making theory and practice: An overview of user interests in knowledge seeking and use, Journal of knowledge management 2 (1998) 36–46.

A. Appendix

A.1. Formative Study Participants

Participant ID	Gender	Education Level	Occupation	Years of LLM Experience	LLM Previously Used
P1	Female	Master's Degree	PhD student	1-2 years	GPT
P2	Male	Doctorate	PhD student	1-2 years	GPT,Claude, Llama, Gemini
P3	Male	Bachelor's Degree	Research Software Engineer	None	None
P4	Male	Bachelor's Degree	Software Engineering Director	1-2 years	GPT, Llama, Other
P5	Female	Master's Degree	Data Engineer	Less than 1 year	GPT, Llama
P6	Male	Master's Degree	Research Software Engineer	1-2 years	GPT, Llama, Gemini

Table 1
Formative participant demographic data

A.2. Data Generation Prompts

You are a patient-intake bot. You will introduce yourself as a physician assistant bot whose role is to ask the patient some questions about their visit. Your role is to ask the user a followup question based on their input in a medically professional manner, one question at a time. Do not make medical recommendations to the user. Input:input

Prompt 2: Prompt to generate outputs as a patient intake bot.

A.3. Prompts Used for LLM-as-a-judge model

You are a helpful assistant that can check the quality of a data point based on a list of provided requirements. You can objectively evaluate the output on the given criteria based on its quality of responses by assessing how well the responses satisfy the given requirements in the criteria and assertion. The requirements will have examples that satisfy (pass) the requirements and examples that fail. You should provide comprehensive feedback on the responses according to the criteria and provide detailed justification for your feedback. If you refer to specific fragments of the responses in your feedback, you should also return these fragments as evidence. You should return your final answer as a valid JSON object of the following format: JSON

Prompt 3: Prompt used to generate evaluation results