Conversational User-AI Intervention: A Study on Prompt Rewriting for Improved LLM Response Generation

Anonymous ACL submission

Abstract

Human-LLM conversations are increasingly becoming more pervasive in peoples' professional and personal lives, yet many users still struggle to elicit helpful responses from LLM Chatbots. One of the reasons for this issue is users' lack of understanding in crafting effective prompts that accurately convey their information needs. Meanwhile, the existence of real-world conversational datasets on the one hand, and the text understanding faculties of LLMs on the other, present a unique opportunity to study this problem, and its potential solutions at scale. Thus, in this paper we present the first LLM-centric study of real human-AI chatbot conversations, focused on investigating aspects in which user queries fall short of expressing information needs, and the potential of using LLMs to rewrite suboptimal user prompts. Our findings demonstrate that rephrasing ineffective prompts can elicit better responses from a conversational system, while preserving the user's original intent. Notably, the performance of rewrites improve in longer conversations, where contextual inferences about user needs can be made more accurately. Additionally, we observe that LLMs often need to – and inherently do – make *plausible* assumptions about a user's intentions and goals when interpreting prompts. Our findings largely hold true across conversational domains, user intents, and LLMs of varying sizes and families, indicating the promise of using prompt rewriting as a solution for better human-AI interactions.

1 Introduction

Many technologies we've come to rely on in our daily lives—from search engines to cellphones now include a component that enables a user to engage with an LLM in the colloquial "chat" format. These highly capable models have unlocked new frontiers in conversational agents and automated reasoning (Liu et al., 2024; Tian et al., 2024). However, users often find it difficult to obtain satisfactory responses from these systems (Wang et al., 2024), or understand how their prompt resulted in a particular response from the LLM (Khurana et al., 2024). A recent study (Babe et al., 2024) involving students writing code-generation prompts with an LLM assistant revealed that the success of a prompt came down to luck—while some prompts proved effective for some models, students with similar Python expertise found it challenging to write prompts that worked consistently.

There could be multiple reasons for a user query receiving an unsatisfactory response. For example, the response might be incorrect, irrelevant, or contain fabrications (Li et al., 2024b; Yehuda et al., 2024; Xu et al., 2024). Other failure cases stem from users having unfounded expectations in the capabilities of AI systems. For example, the user might not know that ChatGPT can't perform certain actions, such as taking a screenshot. The user might also be dissatisfied with an AI abstaining from answering a query that might violate its guidelines (for example, "watch wicked online for free").

Existing tools for designing better prompts are mostly geared towards professionals and NLP practitioners (Zamfirescu-Pereira et al., 2023; Schnabel and Neville, 2024). Meanwhile, most users of commercial chatbots are laypeople, who may not have an intuitive understanding of crafting effective prompts. In fact, Poole-Dayan et al. (2024) recently reported that undesirable LLM behavior disproportionately affects users with lower English proficiency and lower education levels. For more equitable solutions and to engage a wider user base, it is imperative that LLMs deployed as chatbots better interpret users' information needs, in whatever forms they are expressed.

The first step towards developing better solutions is to understand user-AI interaction failure at scale, and to study the impact of potential remediation strategies on real-world conversations. The availability of datasets of human-LLM conversations such as WildChat (Zhao et al., 2024) on the one hand, and the capability of modern LLM systems to analyze, interpret, and rewrite text at scale on the other, presents an opportunity to inform future strategies on improving user-AI interactions. While prior work has leveraged conversational logs to measure user satisfaction (Lin et al., 2024), generate taxonomical categories (Wan et al., 2024), and perform alignment (Shi et al., 2024), no research effort has focused on the impact of sub-optimal prompts on unsatisfactory user outcomes.

Thus in this paper, we investigate the feasibility of rewriting user prompts with LLMs, in ways that **better express their information needs**, and the impact that these rewrites have on LLM-generated response quality. Our investigative framework involves two LLMs: the first is the one the user is having a conversation with, – which we refer to as the `chatbot` – and the second is the one we use to rewrite prompts – which we refer to as the `rewriter`. Given a conversational history between the user and `chatbot`, we study whether the `rewriter` is capable of inferring the information needs of the user, and reformulating a prompt that better captures these needs. We also measure the impact of these rewrites on downstream usability by prompting the `chatbot` to generate a response to the reformulated prompt.

During the process of performing a prompt rewrite, we also ask the `rewriter` to generate additional insights. These include the degree of modification required, the aspects of improvement (such as clarity, or specificity), and the assumptions, if any, the model needs to make in order in to construct an effective prompt[1]. The insights not only serve as a chain-of-thought for the model as it reformulates a user prompt, but provide novel axes along which to analyze user-AI conversations, and the impact of performing strategic interventions.

We apply our investigative framework on a subset of the WildChat dataset consisting of conversations with unsatisfactory user outcomes. Across five pairs of conversational domains and user intents, and leveraging five different LLMs of varying sizes and from different open- and closed-source model families, we demonstrate that LLMs – including smaller ones – are effective prompt rewriters, and that the resulting responses from chatbots are consistently and significantly better. Our experimental results are consistent with both `gpt-4o` as an automatic evaluator, as well as with human judges. Additionally, we find that longer conversations result in better prompt rewrites, aspects of improvement are partly shared, partly diverge across domains, and that models make *plausible* assumptions while rewriting.[2]

2 Preliminaries

We begin by formalizing our problem space, and the dataset we use for our investigative framework.

2.1 Problem Setup

To study if contextually intervening and rewriting human prompts with LLMs can be helpful to response quality, we simulate its effectiveness retroactively[3]. That is, we analyze real-world historical human-LLM conversations and rewrite user prompts at key turns evidencing user dissatisfaction to show those rewrites result in better responses. Admittedly user dissatisfaction does not uniquely stem from sub-optimal prompts; unfounded user expectations, poor LLM responses, abstentions due to safety policies could all be contributing factors. Therefore we design our investigative framework to be robust to these different types of conversational outcomes: rewrites should help with conversations that benefit from prompt reformulation, while broadly maintaining intent and not degrading performance on other types of conversations[4].

Our problem setup is illustrated in Figure 1. Broadly, working backwards from a user turn that evidences dissatisfaction (*DSAT*), we use a `rewriter` LLM to reformulate the preceding user turn, and measure how a `chatbot` LLM responds to this rewritten prompt. In Figure 1 a vaguely written user prompts ("Ruleta Casino") becomes a candidate for a prompt rewrite since it results in a response that dissatisfies the user.

More formally, consider a conversation $C = \{u_1, m_1, ...u_n, m_n\}$ consisting of alternating user turns u_i and model responses m_i, where i is the index of a dialog turn. Moreover, define `chatbot` to be an autoregressive LLM that responds to a user input at turn i: $m_i = LLM_{chatbot}(u_i, H_i; \theta)$,

[1] Sometimes, user prompts are so underspecified that it is impossible to infer underlying needs without making assumptions

[2] We will release all code and data upon acceptance.

[3] The ideal setup to test the helpfulness of interventions would require in-situ A/B testing, which is beyond the scope of our work

[4] Our results (Section 4) demonstrate the general success of this approach, and we discuss other cases in Section 4.4

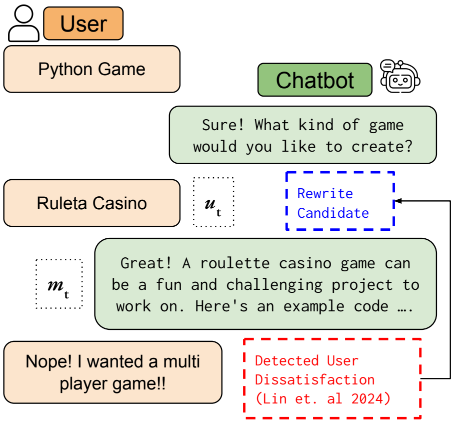
User
Python Game
Chatbot
Sure! What kind of game would you like to create?
Ruleta Casino
u_t
Rewrite Candidate
m_t
Great! A roulette casino game can be a fun and challenging project to work on. Here's an example code ….
Nope! I wanted a multi player game!!
Detected User Dissatisfaction (Lin et. al 2024)

Figure 1: Figure showing how candidate turns are selected for a rewrite. We work backward from a user response expressing dissatisfaction to the query that caused the most recent model response.

where θ are the set of model parameters, $H_i = (u_1, m_1, \dots, u_{i-1}, m_{i-1})$ refers to the conversational history up to user turn u_i. Also, assume `rewriter` to be a (potentially different) LLM that rewrites an existing user prompt u_i: $u'_i = LLM_{rewriter}(u_i, H_i, P; \theta')$, where P is a prompt template (Prompt A.1 in Appendix) the model uses to perform the rewrite.

Then, given a turn u_d in a conversation C that shows evidence of *DSAT*, our problem becomes that of generating:

$$u'_{d-1} = LLM_{rewriter}(u_{d-1}, H_{d-1}, P; \theta') \quad (1)$$

$$m'_{d-1} = LLM_{chatbot}(u'_{d-1}, H_i; \theta) \quad (2)$$

such that $Q(m'_{d-1}) \geq Q(m_{d-1})$ by some quality measure Q.

2.2 Dataset

In order to study this retroactive rewrite setup, we need—(a) A corpus of real-world user-LLM conversations; and (b) Labels indicating which turns result in user dissatisfaction. For (a), we use a subset of WildChat consisting of non-toxic English conversations that have three or more turns. We follow the data setup of Shi et al. (2024), who previously leveraged this subset. Meanwhile, for (b) we use the user satisfaction rubrics proposed by Lin et al. (2024) and adapted by Shi et al. (2024) to assign a label of *SAT*, *DSAT* or *NONE* to every turn in our dataset, and retain those conversations with at least on *DSAT* label.

Our sample of the Wildchat dataset is still very large, and comprises chat interactions covering a wide variety of conversational domains and expressing a range of user intents. To further focus our study of how user dissatisfaction varies across these axes, we classify conversation turns into domains and intents (Wan et al., 2024). Domains cover topical categories such as "Software and Web Development" and "Culture and History", while intents refer to the user's conversational goals such as "seeking information" and "creation".

In our analyses, we group conversations jointly over domain and intent to categorize them with similar information goals. We perform all our analyses for the five most commonly-occurring categories in our dataset, which can be found in Table 1.

Domain	Intent	#Convs	#>=5 Turns	Rewrite Index
Software/Web Dev	Seek Info	2397	446	2.71
Software/Web	Create	1459	197	2.22
Writing/Journalism	Create	376	64	2.39
Tech	Seek Info	349	78	3.49
Math/Logic	Seek Info	346	79	3.30

Table 1: Conversation metrics broken down by Domain and Intent. Rewrite index refers to the average turn corresponding to a candidate rewrite.

3 Conversational Intervention through Prompt Rewriting

Given our goal of using LLMs to perform strategic rewrites, we now describe our approach to instructing them to do so. Given the cost of fine-tuning LLMs, and lack of relevant prompt rewriting data, we instead use a prompting strategy to steer models to our goal. Broadly our approach consists of two instruction categories: the first deals directly with the model performing the rewrite, while the second seeks to generate additional insights that serve as a chain-of-thought (Wei et al., 2022) for better reasoning, as well as novel axes of analysis in our investigative framework. These two categories are illustrated in Figure 2 and are detailed below in Sections 3.1 and 3.2. Our full prompt is included in Appendix A.1.

3.1 Performing Rewrites

Given a candidate turn u_t we first instruct models to reason about the degree which it needs a rewrite on a 3-point scale – `NO MOD` indicating that the `rewriter` has judged the prompt to be adequate, `SOME MOD`, and `HEAVY MOD`. This is to make our pipeline robust to cases that would otherwise not benefit from prompt rewriting.

Then for the cases identified as `SOME MOD` or `HEAVY MOD` we instruct models to generate a better, rewritten prompt u'_t while *maintaining user intent*, according to Equation 1. Using this rewrit-

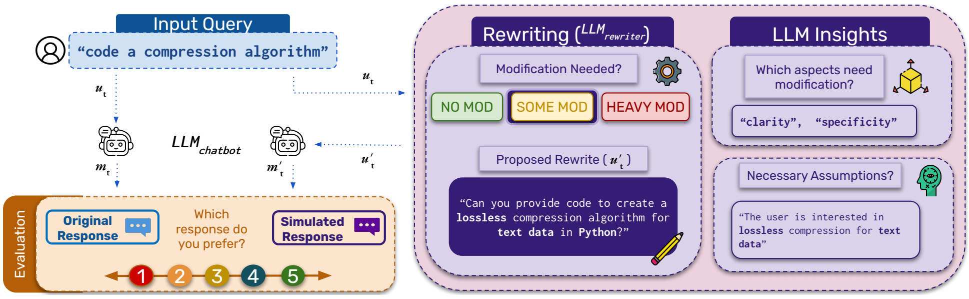
Input Query
“code a compression algorithm”
u_t
u_t
$LLM_{chatbot}$
m_t
m'_t
u'_t
Evaluation
Original Response
Which response do you prefer?
Simulated Response
1
2
3
4
5
Rewriting ($^{LLM}{}_{rewriter}$)
Modification Needed?
NO MOD
SOME MOD
HEAVY MOD
Proposed Rewrite (u'_t)
“Can you provide code to create a lossless compression algorithm for text data in Python?”
LLM Insights
Which aspects need modification?
“clarity”, “specificity”
Necessary Assumptions?
“The user is interested in lossless compression for text data”

Figure 2: This figure summarizes our overall approach to prompt rewriting and evaluation. First, the `rewriter` processes an input user prompt along with the conversational history H with the `chatbot`, and reasons about the aspects in which the query needs to improve, as well as the assumptions needed to make a rewrite before proposing a rewrite (Section 3). Then we comparatively measure the quality of the response to this rewritten prompt against the original, using either an LLM or a human as a judge (Section 4).

ten prompt, we can then generate a new candidate `chatbot` response m'_t using Equation 2.

3.2 Generating Additional Insights

In addition to a reformulated prompt, we also instruct models to reason in a fine-grained manner in order to generate additional insights about the rewriting operation.

Aspects of query improvement. The first category of insights concerns *aspects* – these are open-ended free-text categories such as clarity, specificity, tone, etc. that models are instructed to list as they perform a rewrite. An example with relevant aspects is shown in Figure 2. We use these aspects to understand along what dimensions sub-optimal user queries fall short, as well as gain insights into the ways LLMs perform rewriting operations.

Gathering Model Assumptions The second category of insights focuses on *assumptions* that the model needs to make in order to effectively rewrite a better prompt. This becomes especially relevant in cases when the user input is underspecified, or the historical conversation context lacks sufficient grounding information. In such cases we instruct the model to list *plausible* assumptions about the user's information goals. Figure 2 contains an example of an assumption the model makes while rewriting the input query. Assumptions are useful for understanding how models reason about user needs, as well as measuring the impact that these assumptions have on LLM response quality.

4 Results

As outlined in Section 3 we prompt a model to rewrite a user prompt, and then generate a new LLM response according to Equations 1 and 2. Before presenting our experiments and results we summarize how pairs of original and candidate responses m_t and m'_t are evaluated.

4.1 Evaluating Simulated Responses

Evaluating whether a simulated AI response is more helpful to the user is challenging, as we don't have access to ground truth labels, or the original users. However, we can still evaluate the modified response m'_t on the basis of metrics such as relevance and contextuality, similar to Kwan et al. (2024). Here, either an LLM or a human judge is instructed to carefully consider the conversational history H along with two possible ending turns, – the default ending (u_t, m_t) and the simulated ending (u'_t, m'_t) – then asked to make a judgment of which one is better on a 5-point likert scale. We randomize the order in which model responses appear in the evaluation to account for order effects, and perform this evaluation using a carefully constructed prompt (Prompt A.3 in Appendix).

Using LLMs with judicious prompting for evaluation has become common practice, having been applied successfully to a wide range of tasks (Zheng et al., 2023; Jain et al., 2023; Koutcheme et al., 2024). In this paper our main results are based on using an LLM-as-a-judge (specifically `gpt-4o` (Zhang et al., 2023)), although we also perform human validation (see Section 4.3) to support our findings.

4.2 Key Findings

We apply our framework to studying the contextual rewriting capabilites of five LLMs that constitute a variety of model sizes, families, and both closed- and open-source releases. These are `gpt-4o`, `gpt-4o-mini`, `llama-3-70B-Instruct`, `llama-3-8B-Instruct` (Grattafiori et al., 2024),

		gpt-4o			gpt4o-mini			llama-70B			llama-3-8B			Ministral-3B		
Domain	**Intent**	**W**	**L**	**T**	**W**	**L**	**T**	**W**	**L**	**T**	**W**	**L**	**T**	**W**	**L**	**T**
Software/Web Dev	Seek Info	80.1	19.0	1.0	71.8	27.8	0.4	58.9	40.6	0.5	33.8	66.0	0.2	41.8	57.8	0.4
Software/Web Dev	Create	78.8	19.5	1.7	68.7	29.6	1.7	50.8	47.8	1.4	31.7	67.8	0.5	43.4	55.5	1.1
Writing/Journalism	Create	73.3	25.2	1.5	72.6	27.4	0.0	62.4	36.2	1.4	46.7	53.3	0.0	46.5	53.1	0.4
Technology	Seek Info	86.8	13.2	0.0	79.5	20.5	0.0	74.3	25.7	0.0	55.8	44.2	0.0	43.1	56.6	0.4
Math/Logic	Seek Info	79.5	18.0	2.6	80.0	18.5	1.5	66.9	30.3	2.8	39.9	57.8	2.2	40.8	58.4	0.8

Table 2: Outcomes of rewriting candidate prompts across the top five domain-intent pairs (in decreasing order of frequency in the dataset). gpt-4o scores the original response against the simulated response to the rewrite on a 1-5 Likert scale. A W (or Win) refers to a Likert score of 4 or 5, an L (or Loss) refers to a score of 1 or 2, and T (or Tie) refers to a Likert score of 3.

and Ministral-3B.[5] The results across the five domain-intent pairs in our dataset are presented in Table 2, where we use the same model as rewriter and chatbot.

Rewritten prompts produce better responses from LLMs. For gpt-4o, gpt-4o-mini, and llama-3-70B-Instruct, the proposed rewrites results in better responses overall across multiple domain-intent pairs. In particular, for the "Information Seeking" intent, simulated responses were chosen over the original responses in near or over 80% of cases for both gpt-4o and gpt-4o-mini. The lowest win rates are for writing tasks, although even in these cases the rewrites do result in positive win rates for larger models. We hypothesize that because writing queries are more inherently subjective than math or software-related questions, the resulting rewrites may capture the user's information needs as well as they do in other domains.

While the smaller llama-3-8B-Instruct and Ministral-3B models do not show positive win rates in Table 2, we will demonstrate later in this section that this is due to their weakness as a chatbot, not as a rewriter.

Rewrites are more effective further along in the conversation. One pattern we notice across most models is that proposed prompt rewrites have a better chance of succeeding when rewrites are made deeper into the conversation. Table 3 highlights this finding by separating the performance of models on conversations with fewer than 5 turns, from those with longer conversational histories.

All but the smallest Ministral-3Bmodel demonstrate better performance on longer conversations. Although Ministral-3B supports a 128k context window, it is unable to capture user needs from longer conversations.

Model	**Set**	**Win (%)**	**Loss (%)**	**Tie (%)**
gpt-4o	< 5	78.27	20.79	0.93
	≥ 5	82.97	14.93	2.10
gpt4o-mini	< 5	68.15	31.33	0.51
	≥ 5	81.16	17.44	1.40
llama-3-70B	< 5	56.10	43.44	0.46
	≥ 5	65.78	32.06	2.16
llama-3-8B	< 5	34.52	65.13	0.35
	≥ 5	42.16	57.33	0.51
ministral-3B	< 5	43.93	55.48	0.58
	≥ 5	39.33	59.93	0.74

Table 3: Rewrites deeper into the conversation produce better responses. For all models except Ministral-3B, rewrites deeper into the conversation produce better responses than shallower rewrites.

Nevertheless, the significant gains evidences by all the other models validate our rewrites are truly *contextual*. Further into a conversation, the rewriter has more information about the grounded goals and preferences of the user and is thus able to generate better rewrites, which in turn result in better chatbotresponses.

Smaller models can propose effective rewrites. In the results discussed till this point, the rewriter and chatbot have been the same models.This raises an important question. Is the relatively poor performance of some models caused by their subpar rewriting capabilities, or are they held back in their capability as chatbot s during the response generation stage?

To answer this question, we perform additional experiments where we keep the original rewriterfor our two smallest models, but use gpt-4o as the chatbot responding to the rewrite. Doing this decouples our measurement of the ability of an LLM to understand the user's intent and information needs (as a rewriter), from its ability to respond with relevant information (as a chatbot).

[5] Ministral-3B was released via a blog post.

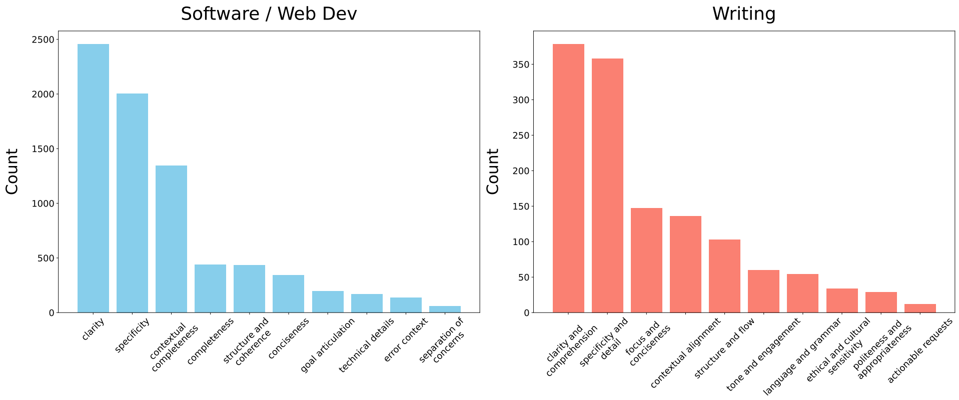

Figure 3: Most common aspects for two of the most common categories in our data - Software and Web Development (Information Seeking) and Writing and Journalism (Creation). The different aspects illustrate how prompts from different categories need improvement along varying dimensions.

Rewriter	Chatbot	Win	Loss	Tie
`gpt-4o`	`gpt-4o`	79.61	19.12	1.27
`llama-3-8B`	`llama-3-8B`	36.49	63.12	0.39
`llama-3-8B`	`gpt-4o`	**58.90**	**40.46**	**0.64**
`Ministral-3B`	`Ministral-3B`	42.63	56.74	0.63
`Ministral-3B`	`gpt-4o`	**66.79**	**31.75**	**1.46**

Table 4: Using a smaller model as `rewriter` with a larger model as the `chatbot`can greatly improve the quality of LLM responses. Results from `gpt-4o` as both are shown in the top row as an upper bound.

The results of this evaluation are presented in Table 4. With `gpt-4o` as the `chatbot` responding to rewritten query results, we observe a more than 20-point jump in the helpfulness of the new responses for both `llama-3-8B-Instruct` and `Ministral-3B`, when they are compared with the original responses. Interestingly, `Ministral-3B` proves to be an even better `rewriter` than `llama-3-8B-Instruct`, based on this finding. While part of these gains are due to the strength of `gpt-4o`as a chatbot, they also demonstrate that smaller models can be effective rewriters. Crucially, what were losing head-to-head comparisons become winning scenarios. This has important implications – when a user prompt is ill-formed and fails to convey their information needs properly, a smaller (even on-device) model might be sufficient to make a rewrite that produces a better response.

4.3 Human Validation

In our paper, all automated evaluation is performed by `gpt-4o`. While LLMs are now commonly used in the community as evaluators (Zheng et al., 2023; Jain et al., 2023), we perform additional human evaluations to ensure the veracity of our findings. Specifically, we instruct human annotators to comparatively evaluate a subset of 100 rewritten responses using the same criteria and 5-point likert scale detailed in Section 4.1.

Validation of `gpt-4o` scores. Five annotators, who are authors of the paper, annotated 40 conversations each, ensuring that each conversation in this subset was annotated by at least two human judges. The score obtained by a rewritten response is calculated as the average of the two annotator ratings it received.

Over this subset, the scores assigned by models and humans are *distributionally different* – `gpt-4o` was much more likely to assign an extreme score. In fact, in 82% of cases it scored a (1) (response 1 is much better) or (5) (response 2 is much better), and zero scores of (3) (both responses are equally good or bad).

In contrast, humans were often ambivalent (23% of cases were assigned a score of 3), and were less frequently extreme (only 12% received scores of 5 or 1). Part of the ambivalence of the human judges was due to the difficulties of making domain-specific judgments. For example, given two pieces of code that are candidate responses to a user query, it is often next to impossible to judge which one is better by purely looking at it. For these reasons, when computing inter-rater agreement between humans and `gpt-4o`, we dropped the conversations receiving a score of 3 from humans, and coarsened our 1-5 scale into a binary label following (Srikanth and Li, 2021) – responses receiving scores less than three (loss), and those receiving a score greater than three (win). Over these samples, human and `gpt-4o`-based judgments received a Krippendorff's alpha of 0.61, showing moderate agreement. Thus, despite the calibration issues resulting from the well-known phenomenon of models exhibiting affinity for extreme distributions (Zhang et al., 2024), we can confirm that gpt-4o based evaluations are reasonably aligned with human judgments.

Validation of intent. In addition to evaluation response quality, we also measure to what extent intents are preserved in prompt rewrites. To do this, on the same subset of 100 conversations, we reveal to annotators which of the two endings is the original and which is the rewritten prompt. Then we ask them to rate on a 3-point Likert scale the degree to which intent is maintained through the rewrite operation. Similar to the previous validation setup, we average the two human-assigned scores.

After averaging, 74% proposed rewrites received a score of 2.5 or 3-indicating strong maintenance of intent, and 21% rewrites received a score of 2, indicating the intent being maintained "somewhat", and only 5% rewrites received a score less than 2. Overall, this illustrates that our rewrites are overwhelmingly **intent-preserving**. We qualitatively discuss some of the cases where user intent was not maintained in what follows.

4.4 Additional Insights

Recall that our investigative framework (Section 3) generates reasoning insights, in addition to performing prompt rewrites and simulating responses. We analyze these insights and what they reveal about the nature of conversational intervention.

Aspects of rewrites vary across domains. Our rewrites yields *aspect* categories of improvement such as "clarity", "conciseness" etc. for each conversation. We consolidate this open ended list of categories iteratively using a modified version of Shah et al. (2024)'s taxonomy induction approach. In Figure 3, we show a list of these consolidated aspects for two key domains in our dataset - Writing and Software/Dev. While many aspects are shared across the domains, some aspects are indeed domain specific. For example, Software prompts require rewrites that focus on "goal articulation", and "error context"; meanwhile Writing prompts are sometimes categorized as needing rewrites that address "appropriateness", and "ethical and cultural sensitivity".

We also investigate how aspects correlate with intent preservation. Specifically, we inspect the human evaluated conversations that received an average intent maintenance score ≤ 2 (see Section 4.3). We find that for the Software domain, among the top aspects of improvement for low-intent preservation scores were "specificity" and "goal articulation". In contrast, most rewrites that successfully preserved the original intent focused solely on "structure and coherence" and "conciseness" – attributes related to refining the existing prompt without adding new information. However, most of the low intent-preserving rewrites came from the Writing domain. Aspects corresponding to these rewrites mainly involved prompts that were explicit or inappropriate in nature, triggering the rewriter LLM's content moderation policies. In those cases, the rewrite converted the prompt to a more appropriate version, but often in the process completely altered the intent of the user.

rewriter makes plausible assumptions. We perform a similar analysis on the *assumptions* our framework generates as part of its reasoning process. Specifically, Table 5 contains examples of assumptions made by gpt-4o while rewriting queries that received better (Winning) or worse (Losing) scores than the original responses. During the intent maintenance validation task, we also ask our annotators to assign a score indicating how plausible the assumptions made in the rewrite are, if any. Annotators identified possible assumptions in 74 out of 100 conversations and in 65% of those cases, the assumption was considered "very" plausible. However, there was no clear correlation between the plausibility of the assumption and the success of the task. This, combined with our finding rewrites later in the conversation produce better responses indicates that while rewriting prompts earlier in the conversation, an LLM should ask a followup question to ground the conversation rather than guessing about the user's intent.

5 Error Analysis

Although our results demonstrate that rewrites generally result in better responses, even our best model (gpt-4o) produced worse responses in 19% of cases (see Table 2). In a closer look, these cases can be divided into two broad categories. First, there are cases where instead of issuing a rewrite, the rewriter interprets the user prompt as an instruction and responds to it. For example, the rewriter, while faced with a candidate prompt that starts with the word "modify: [user input]",

	Winning Assumptions	Losing Assumptions
Software Dev	- The user needs the code to also have the ability to train the model if it is not trained. - The user needs information on syntax and use cases. - The user wants a step-by-step guide.	- User wants to store results in a pandas DataFrame. - User's version of Excel is 2016. - The user wants general tips to manage git commits more effectively given their programming habits.
Writing	- User wants the story to be continued in the same style and tone. - The user wants an expansion of the existing paragraph rather than a new paragraph entirely. - The user wants the sentence to maintain an analytical and logical tone.	- The user wants the story to follow the percentage breakdown of the four-act structure. - The user wants the story to be interesting or humorous, rather than inappropriate. - The last sentence provided is the one needing revision.

Table 5: Winning and Losing Assumptions for Software Development and Writing Contexts.

directly modifies the input instead of reformulating the prompt. These are cases where the model fails to follow the rewriting instructions.

The second category are cases where the original user prompt is trying to jailbreak the safety guidelines of an LLM. For the Software domain, these may contain prompts that ask to write code that perform an illegal activity, such as obtaining secret keys from a website. For Writing, these mainly involve cases where the user requests for content that is inappropriate or explicit in nature. These findings highlight the need for future solutions that focus on conversational intervention, to balance safety with user intent preservation.

6 Related Work

While our focus on using LLMs to rewrite queries in human-LLM conversations is relatively new, researchers have explored the potential of rewriting SQL queries with LLMs in the recent past (Liu and Mozafari, 2024). In a different setting Ma et al. (2023) used a Retrieval-Augmented LLM to rewrite questions in a single-turn QA setting.

Li et al. (2024a) perform prompt-rewriting using a combination of supervised and reinforcement learning. However, their work focused solely on single-turn document generation tasks, not multi-turn conversations. Their work also differs in requiring to update model weights, which might not always be feasible with limited compute. In a recent effort towards understanding implicit intent in user queries, Qian et al. (2024) propose a dataset and a benchmark for evaluating model understanding of vague or underspecified user prompts. Although their work does not involve rewriting user prompts, they show that with further training, LLMs can be made better at recovering plausible details missing in the user prompt.

Related to our evaluation setup, Malaviya et al. (2024) include synthetically generated contexts to help the LLM make better judgments while choosing one response over the other. However, since our setup leverages real-world conversational histories, we do not need synthetic contexts.

7 Conclusion and Future Work

State-of-the-art LLMs have never been more accessible for completing everyday tasks. Yet, for a significant number of people, LLM responses continue to remain dissatisfactory (Poole-Dayan et al., 2024). In this paper we propose an investigative framework that studies the capability of LLMs to provide contextual interventions in human-AI conversations. Specifically we design an LLM-centric process that operates over a conversational history to rewrite an input prompt, while generating novel reasoning insights. Our experiments based on both human and LLM evaluation demonstrate that responses to these rewritten prompts are consistently better, and that even smaller LLMs prove to be effective prompt rewriters. We also show how longer histories with greater conversational context lead to better prompt rewrites. Finally we perform detailed analyses on the reasoning insights yielded by our rewriting framework: we note how aspects vary across domains, and how *plausible* model assumptions correlate with better responses.

Our findings from this novel study of LLM-based prompt rewriting, and LLM-based prompt rewriting as an in-situ remediation strategy has implications on the design and deployment of future AI chatbot systems. First, LLMs – even small, on-device ones – could be used effectively as prompt rewriters, although larger models are still required to offer better responses. Second, models need to become better at making plausible, grounded assumptions about users, while asking good clarifying questions when that grounding is insufficient. Third, LLMs need to improve their long-context understanding and reasoning capabilities, since these often correlate with better outcomes for users. We leave these directions to future work.

Limitations

While we show that LLMs can make effective rewrites, it is quite challenging to evaluate their helpfulness without deploying them in an in-situ setting. For writing tasks, human preferences for tone, style or brevity could be extremely subjective. For coding tasks in general, evaluating LLM responses require domain experts, or a controlled setting where code can be compiled and tested for correctness for real-world tasks. We propose a general setting, involving various domains and intents, but future work might hone in on a particular domain and gather experts to perform domain-specific evaluations. While none of our rewrites aided a jailbreak attempt, it is possible that the LLM rewrites such an attempt without thwarting it.

References

Hannah Babe, Sydney Nguyen, Yangtian Zi, Arjun Guha, Molly Feldman, and Carolyn Anderson. 2024. StudentEval: A benchmark of student-written prompts for large language models of code. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8452–8474, Bangkok, Thailand. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, and Anirudh Goyal et. al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. *arXiv preprint arXiv:2306.01200*.

Anjali Khurana, Hariharan Subramonyam, and Parmit K Chilana. 2024. Why and when llm-based assistants can go wrong: Investigating the effectiveness of prompt-based interactions for software help-seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, page 288–303, New York, NY, USA. Association for Computing Machinery.

Charles Koutcheme, Nicola Dainese, Sami Sarsa, Arto Hellas, Juho Leinonen, and Paul Denny. 2024. Open source language models can provide feedback: Evaluating llms' ability to help students using gpt-4-as-a-judge. *Preprint*, arXiv:2405.05253.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024. MT-eval: A multi-turn capabilities evaluation benchmark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20153–20177, Miami, Florida, USA. Association for Computational Linguistics.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2024a. Learning to rewrite prompts for personalized text generation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 3367–3378. ACM.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024b. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Ying-Chun Lin, Jennifer Neville, Jack W Stokes, Longqi Yang, Tara Safavi, Mengting Wan, Scott Counts, Siddharth Suri, Reid Andersen, Xiaofeng Xu, et al. 2024. Interpretable user satisfaction estimation for conversational systems with large language models. *arXiv preprint arXiv:2403.12388*.

Jie Liu and Barzan Mozafari. 2024. Query rewriting via large language models. *Preprint*, arXiv:2403.09060.

Na Liu, Liangyu Chen, Xiaoyu Tian, Wei Zou, Kaijiang Chen, and Ming Cui. 2024. From llm to conversational agent: A memory enhanced architecture with fine-tuning of large language models. *Preprint*, arXiv:2401.02777.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Chaitanya Malaviya, Joseph Chee Chang, Dan Roth, Mohit Iyyer, Mark Yatskar, and Kyle Lo. 2024. Contextualized evaluations: Taking the guesswork out of language model evaluations. *Preprint*, arXiv:2411.07237.

Elinor Poole-Dayan, Deb Roy, and Jad Kabbara. 2024. Llm targeted underperformance disproportionately impacts vulnerable users. *Preprint*, arXiv:2406.17737.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1088–1113, Bangkok, Thailand. Association for Computational Linguistics.

Tobias Schnabel and Jennifer Neville. 2024. Prompts as programs: A structure-aware approach to efficient compile-time prompt optimization. *arXiv preprint arXiv:2404.02319*.

Chirag Shah, Ryen W. White, Reid Andersen, Georg Buscher, Scott Counts, Sarkar Snigdha Sarathi Das, Ali Montazer, Sathish Manivannan, Jennifer Neville, Xiaochuan Ni, Nagu Rangan, Tara Safavi, Siddharth Suri, Mengting Wan, Leijie Wang, and Longqi Yang. 2024. Using large language models to generate, validate, and apply user intent taxonomies. *Preprint*, arXiv:2309.13063.

Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, Zexue He, Mengting Wan, Pei Zhou, Sujay Jauhar, Xiaofeng Xu, Xia Song, and Jennifer Neville. 2024. Wildfeedback: Aligning llms with in-situ user interactions and feedback. *Preprint*, arXiv:2408.15549.

Neha Srikanth and Junyi Jessy Li. 2021. Elaborative simplification: Content addition and explanation generation in text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5123–5137, Online. Association for Computational Linguistics.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17659–17681, Miami, Florida, USA. Association for Computational Linguistics.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W White, Longqi Yang, et al. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5836–5847.

Jiayin Wang, Weizhi Ma, Peijie Sun, Min Zhang, and Jian-Yun Nie. 2024. Understanding user experience in large language model interactions. *Preprint*, arXiv:2401.08329.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Yakir Yehuda, Itzik Malkiel, Oren Barkan, Jonathan Weill, Royi Ronen, and Noam Koenigstein. 2024. InterrogateLLM: Zero-resource hallucination detection in LLM-generated answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9333–9347, Bangkok, Thailand. Association for Computational Linguistics.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. 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*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, Zico Kolter, and Daphne Ippolito. 2024. Forcing diffuse distributions out of language models. *Preprint*, arXiv:2404.10859.

Yue Zhang, Ming Zhang, Haipeng Yuan, Shichun Liu, Yongyao Shi, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Llmeval: A preliminary study on how to evaluate large language models. *Preprint*, arXiv:2312.07398.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Appendix

A.1 Prompts

In this section, we outline the prompts used for rewriting user prompts and for comparing model responses. During human annotation of the 100-conversation subset, the annotators were shown a condensed version of Prompt A.3.

Prompt A.1: Rewriting User Prompts

Prompt: **Goal**: Given a user's **query** and their conversational history with an AI Chatbot, your task is to identify the aspects in which the **query** can be improved or if it's already optimal, identify the aspects in which it is already effective. To do so, first analyze the query for aspects of improvement or describe aspects that are already effective. Then, propose a list of one or more possible rewrites that communicates the user's needs and goals more effectively as an input to an AI Chatbot while keeping the user intent intact. Be careful not to change the goal or the intent of the user when you propose a rewrite keeping in mind the **Conversational History**. For each rewrite, if you have to add any new information that is not present in the **Conversational History** to make the query better, list the assumptions you need to make.

Task: Given a user **Query**, your task is to output the following:
First, output whether or not the **Query** needs modification for eliciting an effective response from an AI Chatbot. If it's a good query and doesn't need any modification at all, output NO MOD. If it requires some modification, output SOME MOD. If the **Query** requires to be heavily rewritten, output HEAVY MOD.

If you chose NO MOD, output the aspects of the **Query** that makes it an effective query in a markdown table in the following format:
<table format>
If the query needs any rewrite (that is, if you answered SOME MOD or HEAVY MOD in the previous question), output the aspects of improvement in a markdown table in the format below:
<table format>
DO NOT answer the input **Query**, your job is only to evaluate how well it expresses the user's information need from a Chatbot.

Conversational History: query_context
Query: target_query

Prompt A.2: Rewrite Output Format

Continued from A.1: If you propose a list of rewrites, then for each rewritten query, list the following information:
Rewrite: <The Rewritten Query. Make sure to include ALL relevant information from the original **Query** and the **Conversational History**>

Information Added: <Whether information beyond what's present in the **Query** or the **Conversational History** needs to be added in the rewrite. Reply YES or NO>
Assumptions: If there's additional information needed to be added to the user's query for it to be effective, then those are assumptions about the user's goals that need to be made. If you answered YES in the previous step, list the assumptions along with how salient they are for the rewrite, and how plausible they are for the user to believe in from a scale of HIGH, MID and LOW in a markdown table in the format below:
|assumption|salience|plausibility|
|<assumption text>|<HIGH, MID or LOW>|<HIGH, MID or LOW>|
Note:
The conversational history may or may not be present, and it provides you with some context on the user query you need to analyze. If the context is about a different task or topic, discard it.
Order the rewrites from the most likely to the least.

Output using the template outlined below:
<START OF OUTPUT TEMPLATE>:
...
<END OF OUTPUT TEMPLATE>
Conversational History: query_context
Query: target_query
Based on the **Query** and the **Conversational History**, fill out the OUTPUT TEMPLATE in order to structurally analyze the user **Query** in context without trying to answer the query.

Prompt A.3: Evaluation Prompt

Prompt: **Context**
You will be shown conversations of a user with a chatbot providing users with information about their queries in their preferred language based on its knowledge, common sense and natural language understanding. The chatbot task is to infer the user's intent and context from the conversation and provide the most relevant and useful information to the user.
Instructions
You are given a conversational history between a user and the chatbot, and two possible endings to that conversation. From the conversational history, you need to understand the goals of the user in the conversation and determine which of the two endings better satisfy the information needs of the user.
Scoring Guidelines
Make a judgment on which of the two endings offers a richer user experience on a 5-point Likert scale, where:

1 indicates that the model response in Ending 1 is **much better** than Ending 2
2 indicates that the model response in Ending 1 is **somewhat better** than Ending 2
3 indicates that the model response in Ending 1 and Ending 2 are equally good/bad
4 indicates that the model response in Ending 2 is **somewhat better** than Ending 1
5 indicates that the model response in Ending 2 is **much better** than Ending 1

General evaluation guidelines

The response must be **contextual**, taking all conversational context into account while responding to user.
The response should be directly **relevant** to the user's query and be essential to satisfy their information needs.

A.2 Annotation Instructions for Intent Preservation

Prompt A.4: Intent Preservation Annotation Instruction

Instruction: **Question 1:** In this task, you will be provided the conversational history between the user and the chatbot. This time, it will be revealed which ending is the Original Ending and which is the Rewritten one.
Your task is to answer two questions about the original vs rewritten prompt (without considering the model responses) on a 3-point Likert scale.
Task:
Question 1: To what extent is the intent of the user as expressed in the original ending and the conversational history carried over in the rewrite? Return a score from 1 to 3, where

1 indicates that the rewritten prompt does not at all maintain the same intent as the original prompt.
2 indicates that the rewritten prompt somewhat maintains the overall intent of the original prompt.
3 indicates that the rewritten prompt maintains the exact same intent as the original prompt.

Prompt A.5: Assumption Plausibility Annotation Instruction

Instruction: **Question 2:** If the rewrite does change the intent (you chose 1 or 2 in the previous step), are there any assumptions made by the model in constructing the rewrite? If so, assign a score of 1-3 that denotes how plausible the assumptions are in satisfying the information goals of the user, given the intent expressed in the conversational history and the original prompt.
Assign a score from 1-3 where:

1 indicates that the assumptions are not plausible at all, given the intent described by the user in the available data.
2 indicates that the assumptions are somewhat plausible given the intent described by the user in the available data.
3 indicates that the assumptions are very plausible given the intent described by the user in the available data.

If there are no assumptions made by the model, leave this section blank.

A.3 Prompting Parameters

For all our rewriting and simulation experiments, we use a temperature of 1. For evaluation, the temperature is always set to zero.