

Reading Between the Turns: User Repair as a Context-Dependent Signal of Breakdown and Recovery in Multi-Turn LLM Assistants

Leah Michel^{1,2}, Nuno Miguel Martins Pacheco¹, Choro Ulan uulu¹, Mikhail Kulyabin¹, Jan Joosten¹, Filippas Petridis¹, Michael A. Hedderich^{2,3}, Barbara Plank^{2,3}

¹Siemens AG, Germany, ²MaiNLP lab, LMU Munich, Germany,

³Munich Center for Machine Learning, Germany

Correspondence: leah.michel@siemens.com

Abstract

In multi-turn interactions with deployed LLM assistants, users rarely provide explicit feedback, making it difficult to assess whether the interaction succeeded or failed. We therefore investigate user repair: follow-up user actions such as rephrasing, correcting, clarifying, or requesting more information. Such actions may signal breakdown recovery, but they can also reflect ordinary task progression, making them difficult to interpret in isolation. Using a human-annotated dataset of industrial multi-turn conversations, we examine when repair is informative about assistant-side deficiencies and user dissatisfaction. We find that repair occurs more often after deficient responses, but it is also common after non-deficient responses, making it a weak standalone signal of failure. Its diagnostic value becomes clearer when it is interpreted jointly with assistant response quality, repair type, and dialogue context. Repetition/rephrasing is most strongly associated with negative feedback on deficient responses, whereas requests for more information are more common after non-deficient responses, suggesting a role in collaborative task progression. We further show that repair is not always local: while most repairs target the immediately preceding turn, a substantial minority target earlier dialogue context. These findings position user repair as a context-dependent interactional signal for monitoring, evaluating, and diagnosing deployed assistants.

1 Introduction

Large language model (LLM)-based assistants are increasingly being deployed in industrial troubleshooting, maintenance, and procedural support scenarios, where they provide access to technical documentation, expert knowledge, and operational data through natural language interaction (Narimani and Klarmann, 2024; Shin et al., 2025; Li et al., 2026). Technicians often engage with these systems during ongoing diagnostic processes

requiring iterative exploration of machine conditions, documentation, and corrective actions. Consequently, interactions are multi-turn, with users refining, clarifying, and extending their requests toward task completion.

Monitoring assistant quality in deployment is challenging because explicit user feedback (e.g., pressing a thumbsDown button) is sparse, and the absence of negative ratings does not imply success. Instead, interaction issues often surface indirectly through user follow-up behavior such as rephrasing, adding information, requesting clarification, or correcting the assistant. However, such behavior is interactionally ambiguous: the same surface form can function as repair or ordinary task progression, depending on the adequacy of the prior assistant response and the broader dialogue context. For example, a user who asks a follow-up question may be continuing a successful troubleshooting process, but the same behavior may also reflect an attempt to recover from an insufficient or misunderstood response.

This raises a central challenge: how should user follow-up behavior be interpreted as evidence of misunderstanding versus successful interaction? We focus on **user repair**, broadly understood as practices for addressing troubles in speaking, hearing, or understanding in interaction (Schegloff et al., 1977; Moore et al., 2024; Mohapatra et al., 2024). In our setting, repair includes behaviors such as rephrasing, clarification, correction, information addition, and requests for more information following assistant responses. These behaviors overlap with clarification mechanisms studied in conversational AI (Rahmani et al., 2023), but their meaning depends on context.

Building on dialogue-based accounts of repair and common-ground maintenance (Clark and Schaefer, 1989; Ginzburg, 2012), we argue that user repair should not be treated as a standalone proxy for assistant failure. We study this using

a human-annotated dataset of multi-turn industrial conversations labeled along three dimensions: assistant-side response deficiencies (e.g., information failure, entity mismatch, or refusal), user response strategies (e.g., repetition/rephrasing, information addition/update, request for more information, clarification request, or correction), and repair-target locality, i.e., whether a repair targets the immediately preceding turn or earlier dialogue context.

We address three research questions: (i) whether repair follows assistant deficiencies (RQ1), (ii) whether and when repair is informative about user satisfaction and dissatisfaction (RQ2), and (iii) whether repair is primarily local to the immediately preceding turn or also targets earlier turns (RQ3).

We find that repair is informative but insufficient on its own. It is more likely after assistant responses annotated as deficient, yet remains common after responses not annotated as deficient (RQ1). In the latter case, requests for more information are especially common, suggesting that some follow-up behavior reflects ordinary task progression or the need to continue a task despite a superficially acceptable response. With respect to user feedback, repair alone is a weak signal, but it becomes more informative when interpreted jointly with assistant-side deficiency, especially for negative feedback and particularly when users repeat or rephrase their requests and the paired assistant response is annotated as deficient; corresponding patterns for positive feedback are weaker (RQ2). We also find that repair is not always local (RQ3): while most repairs target the immediately preceding turn, a substantial minority refers to earlier conversational context. These findings suggest that, on its own, user repair should not be treated as an indicator of failure, but as a context-dependent signal that can support more effective evaluation, monitoring, and recovery strategies for deployed assistants.

This paper makes three contributions:

1. We reconceptualize user repair in multi-turn human-AI interaction not as a direct proxy for misunderstanding, but as a context-dependent interactional signal whose meaning emerges only in relation to assistant behavior and dialogue history.
2. We introduce an annotation framework linking assistant-side deficiencies, user response

strategies, and repair-target locality, and identify interactional configurations (e.g., repetition/rephrasing paired with deficient assistant responses) in which breakdown risk and dissatisfaction become especially observable in user behavior.

3. We provide evidence that repair is not purely turn-local but often discourse-level, with a substantial minority targeting earlier context, highlighting that breakdown and recovery operate across turns rather than within isolated exchanges.

Together, our findings suggest that understanding in human-AI interaction is not directly observable in individual responses, but must be inferred from interactional dynamics across turns, where user repair serves as a context-dependent signal of breakdown risk and recovery.

2 Related Work

Our work connects three strands of research: assistant-side failure analysis, user repair and conversational grounding, and evaluation from implicit interaction signals.

Assistant-side failures and conversational evaluation. Prior work has developed taxonomies of assistant-side failures (Higashinaka et al., 2015, 2021) and evaluation frameworks for multi-turn and goal-oriented assistants (Piskala et al., 2025). In LLM-based systems, related work has examined response-quality failures such as hallucination, unsupported claims, and factual inconsistency (Zhang et al., 2025), as well as broader degradation in multi-turn conversation, including context loss and failure to recover once interaction has gone off track (Laban et al., 2026). These approaches help identify *what went wrong* on the assistant side, but provide less insight into *how assistant-side deficiencies are reflected in subsequent user behavior* and how such behavior can inform evaluation.

User repair and conversational breakdown. A complementary line of work studies clarification requests and repair behaviors as responses to misunderstanding in dialogue (Rodríguez and Schlangen, 2004; Stoyanchev et al., 2013; Ngo et al., 2024). Recent work emphasizes that understanding in conversational interfaces is bidirectional, with users actively repairing agent responses rather than repair being limited to user-input problems (Moore et al.,

2024). Related HCI work frames breakdown recovery as a collaborative process distributed across user and system actions (Gnewuch and Reinke-meier, 2026), while review work highlights fragmented terminology and limited focus on user-initiated repair (Alghamdi et al., 2024). Foundational accounts from conversation analysis and formal dialogue theory further treat repair as a routine mechanism for addressing troubles in interaction and updating common ground (Schegloff et al., 1977; Ginzburg, 2012). Emerging work on LLM-based assistants shows that user-initiated repair is frequent, effortful, and shaped by task context (Mathur et al., 2025). Recent work on human-LLM grounding further shows that early grounding failures predict later interaction breakdown and that LLMs are less likely than humans to initiate clarification or other grounding-supportive moves, reinforcing the need to study breakdown as an interactional rather than purely response-level phenomenon (Shaikh et al., 2025). Across these strands, however, prior work does not always clearly distinguish repair-oriented responses to interactional trouble from broader follow-up behaviors such as requests for more information or elaboration, which may also occur during ordinary task progression.

Implicit feedback and post-deployment signals. Prior work on implicit feedback and post-deployment learning shows that behavioral traces and free-text reactions can provide useful supervision for improving conversational systems, but are often sparse, noisy, and context-dependent (Pang et al., 2024; Park et al., 2021; See and Manning, 2021). This is especially relevant in LLM-based systems, where failures of contextual grounding and shared-reference tracking shape how user follow-up behavior should be interpreted (Shaikh et al., 2024).

Positioning of our work. We address this gap by introducing a joint annotation and analysis framework that links assistant-side deficiencies, user follow-up behavior, and repair-target locality. This allows us to ask when repair and related follow-up behavior signal misunderstanding, grounding difficulty, or unresolved interpretation, versus routine task progression. Rather than treating repair as a direct proxy for failure, we analyze it as a *context-dependent interactional signal* whose diagnostic value emerges only when jointly considered with assistant-side context.

3 Data, Reference Set, Annotation

3.1 Real Data from a Deployed System

The data were derived from logs of a deployed industrial assistant used for troubleshooting, maintenance, and procedural support. The assistant is a retrieval-augmented generation (RAG) system with access to technical documentation. Interactions are typically multi-turn and incremental, with users refining requests by adding details such as error codes, machine context, or observed behavior.

3.2 Reference Set and Unit of Annotation

The full dataset consists of 167 multi-turn sessions gathered between August 2024 and July 2025. From this pool, we selected 62 sessions comprising 359 interaction units for annotation and analysis (see Table 1). The annotated reference set was constructed to maximize coverage of user repair and assistant-side response deficiencies rather than to estimate their prevalence in deployment. Session selection followed a three-bucket strategy: **Bucket A** (32 sessions) contains interactions with a high likelihood of repair and assistant-side deficiencies, identified using heuristics such as lexical cues and explicit user feedback; **Bucket B** (15 sessions) contains borderline cases; and **Bucket C** (15 sessions) contains randomly sampled sessions. This sampling strategy supports fine-grained analysis of these phenomena, but it may amplify their frequency relative to live system behavior. Accordingly, descriptive rates and comparisons reported in this paper should be interpreted within the annotated set rather than as unbiased estimates of deployment-wide frequencies.

Anonymized user identifiers were available, but the present analysis does not model user-level effects. We acknowledge that repeated users and differences in usage style or proficiency may influence repair behavior, and leave this to future work.

Due to industrial privacy and confidentiality constraints, the raw interaction logs cannot be publicly released. To support transparency, we provide the annotation taxonomy, annotation guidelines, aggregate statistics, and de-identified illustrative examples.¹

We define an interaction unit as a user utterance and its corresponding assistant response, (U_t, A_t) , interpreted with access to the preceding turns in the

¹Project materials: <https://github.com/mainlp/reading-between-the-turns>

Statistic	Count	% of units
Annotated interaction units	359	100.0
Sessions	62	—
Trigger units	115	32.0
Units with at least one repair label	157	43.7
Units with thumbsDown	28	7.8
Units with thumbsUp	76	21.2

Table 1: Composition of the human-annotated reference set. Percentages are calculated over annotated interaction units.

same session. Each unit is annotated for assistant-side response deficiency (**trigger**) and user response strategy (**URS**), allowing us to analyze assistant response quality together with the user’s subsequent response in context.

3.3 Assistant-side Deficiency Annotation

We use the term *trigger* as shorthand for an observable assistant-side response deficiency that could plausibly elicit repair or dissatisfaction. A response is *trigger-positive* if it falls under any of the following categories: *information failure*, *information failure + guidance*, *system failure*, *refusal*, *entity mismatch*, and *context loss* (see definitions in Appendix 10). Otherwise, a response is considered *non-trigger*. We emphasize that trigger is not intended as a causal diagnosis of system failure, but as an interaction-facing proxy for response deficiency grounded in user-perceived utility. This distinction allows us to relate assistant behavior to observable interaction dynamics without assuming access to system-internal causes.

Trigger annotation is intentionally local: labels are assigned based on the current interaction step and the preceding turns, rather than with retrospective access to the full future dialogue. As a result, some responses may be annotated as non-trigger at the time they occur, even though later user behavior reveals unresolved contextual trouble. A repair-containing user turn may therefore respond either to the immediately preceding assistant response or to an earlier assistant turn; thus, a repair in U_t does not necessarily imply that A_{t-1} is trigger-positive.

3.4 User Response Strategy Annotation

URS labels describe observable user behavior in follow-up turns after assistant responses (see Appendix A.1). The scheme includes three families: **repair**, **steering**, and **task management**. Repair labels are repetition/rephrasing ($R1$), information addition/update ($R2$), request for more informa-

tion ($R3$), clarification request ($R4$), and explicit correction ($R5$).

Some repair types, especially $R2$ and $R3$, may arise both in breakdown recovery and in ordinary task refinement. $R2$ is defined behaviorally as the user adding or updating task-relevant information, and $R3$ as a follow-up request for additional task-relevant information, without assuming that the prior assistant response was objectively incomplete. This design choice reflects the technical nature of the domain: unless a prior assistant response is an obvious *trigger* (e.g., refusal), annotators cannot always determine reliably whether a superficially acceptable assistant response is in fact insufficient for the user’s situated need. Defining $R2$ and $R3$ behaviorally avoids requiring annotators to infer hidden deficiency from the user’s follow-up alone. Both labels are therefore interactionally ambiguous: they may reflect routine task progression when a user incrementally refines an ongoing task, but they may also function as repair when the user adds missing context or seeks information needed to recover from a response that left relevant context unresolved. For example, a user may ask for additional details about a replacement part after an otherwise usable assistant response, which we treat as routine task progression. By contrast, a later request for the machine’s current state may function as repair when it seeks to recover contextual grounding left unresolved by an earlier response.

$R4$ captures cases where the user signals lack of understanding of meaning, reference, or reasoning.

Steering includes output restriction ($S1$), e.g., requesting that the assistant present its answer in table form, and terminology/language control ($S2$), e.g., asking for the answer in German. Task management is represented by task change ($T1$). These labels cover the cases observed in our data.

URS is multi-label, allowing combined behaviors to be captured. For repair-containing user turns, annotators additionally mark the assistant turn targeted by the repair. In many cases, the target is the immediately preceding assistant response A_{t-1} , but repairs may also target earlier assistant turns when users continue addressing an unresolved problem across multiple turns. From this annotation, we derive **repair-target locality**: a repair is **local** if it targets A_{t-1} , and **non-local** if it targets an earlier assistant turn. We also record **repair-target distance**, defined as the number of assistant turns between the repair-containing user turn and the targeted assistant response; distance 1 corre-

sponds to a local repair, whereas larger distances indicate non-local repair. This matters because a repair-containing user turn may follow a locally non-trigger response while still targeting an earlier assistant turn that introduced the unresolved problem.

3.5 Annotation Procedure and Agreement

Two in-house annotators labeled the data independently, with disagreements resolved by a third annotator. For trigger labels, agreement before adjudication was 0.837 exact agreement and Cohen’s $\kappa = 0.654$. For URS, exact-set agreement was 0.423 and MASI (Passonneau, 2006) agreement was 0.535, where MASI is appropriate for partially overlapping multi-label annotations. For repair-target annotation, exact agreement was 0.758 ($\kappa = 0.701$), and locality-level agreement was 0.885 ($\kappa = 0.754$) (Appendix A.2). This repair-target annotation is central to our analysis because it allows us to distinguish repairs that respond to the immediately preceding assistant turn from those that refer back to earlier unresolved context. As expected, agreement is lower for fine-grained multi-label user behavior than for binary trigger and locality distinctions, reflecting the inherent ambiguity and overlap of naturally occurring repair behavior in real support interactions.

4 Tasks and Methods

We analyze user repair at three complementary levels: adjacency, within-turn outcomes, and repair-target locality. Repair is annotated on user utterances, whereas trigger is annotated on assistant responses. Although many repairs target the immediately preceding assistant response A_{t-1} , repairs may also target earlier assistant turns when users continue addressing an unresolved problem across multiple turns. We therefore distinguish adjacency-based analyses from target-based analyses.

First, we examine whether repair in a user utterance U_t is more frequently observed after a trigger-positive assistant response A_{t-1} , while recognizing that not all repairs are local. Second, we study within-turn outcomes by analyzing how explicit feedback on A_t varies with repair in U_t and trigger status of A_t , assessing whether explicit feedback concentrates in particular combinations of repair type and trigger status within the same interaction step. Third, for repair-containing turns, we analyze repair-target locality and the distribution of repair-

Turn	Utterance and Annotations
A_{t-1}	Assistant: "The provided documents do not contain any information about error 12080." <i>Trigger: TRUE (information failure)</i>
U_t	User: "I mean error 1280 after solder wave." <i>URS: R2 (information addition / update)</i> <i>Repair: TRUE, Target: A_{t-1} (local)</i>
A_t	Assistant: "Error 1280 after solder wave typically indicates a temperature imbalance..." <i>Trigger: FALSE</i>

Example of a local repair. The user utterance U_t contains repair in response to the preceding assistant response A_{t-1} . Within interaction step t , repair annotations are associated with U_t , while trigger annotations are associated with the paired assistant response A_t .

Figure 1: Illustration of the annotation schema for a local-repair case.

target distance to assess whether local responses are sufficient for inferring breakdown and whether repair distance relates to explicit user feedback.

Trigger is annotated on assistant responses, whereas repair is annotated on user utterances and interpreted relative to prior assistant turns. A user utterance U_t is considered *repair-containing* if it includes at least one repair label ($R1$ – $R5$). Because URS is multi-label, an utterance may contain multiple repair actions. In binary analyses, repair is treated as present if any repair label is assigned. Subtype analyses focus on repair types with enough observations to support meaningful comparison. Several repair subtypes, particularly $R4$ and $R5$, occurred too infrequently for separate quantitative analysis. Consequently, subtype findings are driven primarily by the more frequent repair categories, particularly $R1$ and $R3$. First-turn user utterances (U_1) cannot contain repair.

Figure 1 illustrates the annotation schema for a local repair case, including the relation between a repair in U_t and the preceding assistant response A_{t-1} , within-turn pairing ((U_t, A_t)), and the attachment of trigger status. Figure 2 complements this with a non-local case, in which a repair in U_t targets an earlier assistant turn rather than the immediately preceding response.

We also include a secondary exploratory prediction analysis of explicit negative feedback using regularized logistic regression over binary repair and trigger features. We use logistic regression as an interpretable low-capacity model to compare the relative predictive contribution of different feature families, rather than to maximize predictive performance. Because negative feedback is sparse,

Turn	Utterance and Annotations
A_{t-k}	Assistant: "The latest value of problem_outline is: ..." <i>Trigger: FALSE</i>
...	<i>Several intervening turns, including assistant responses that do not resolve the earlier information need.</i>
U_t	User: "Are you aware of problem_outline?" <i>URS: R4 (clarification request)</i> <i>Repair: TRUE, Target: A_{t-k} (non-local)</i>
A_t	Assistant: "I'm not aware of the specific details regarding problem_outline ..." <i>Trigger: TRUE (information failure + guidance)</i>

Example of a non-local repair. The user turn U_t refers back to an earlier assistant turn rather than the immediately preceding response.

Figure 2: Illustration of a non-local repair, where the user’s follow-up refers back to an earlier assistant response.

Preceding assistant response	Repair (n/N)	Rate
Non-trigger	107/217	49.3%
Trigger	50/80	62.5%

Table 2: Repair prevalence in the user utterance U_t , stratified by whether the preceding assistant response A_{t-1} is trigger-positive.

we report PR-AUC as the primary metric and treat this analysis as supplementary evidence only (see Appendix C).

This setup allows us to assess whether assistant-side deficiencies are associated with subsequent repair across turns, whether repair becomes informative about explicit user feedback when interpreted jointly with assistant-side context, and whether local adjacency alone is sufficient to capture the interactional source of breakdown.

5 Results

Repair is associated with assistant-side deficiencies, but is not sufficient on its own. We first examine cross-turn dependencies between consecutive turns ($A_{t-1} \rightarrow U_t$). Repair is more frequent following trigger-positive responses than following non-trigger responses (62.5% vs. 49.3%; Table 2). This corresponds to an odds ratio of 1.72, indicating a moderate association while also showing that repair is common in both conditions. Repair therefore cannot be interpreted as a breakdown signal in isolation.

A binary repair flag shows only whether repair

Repair feature	Non-trigger	Trigger
Any $R1$ -containing repair	54.7%	80.0%
Any $R3$ -containing repair	41.5%	20.0%
$R1$ only	31.1%	46.0%
$R3$ only	29.2%	6.0%

Table 3: Repair composition among repair-containing next-turn user utterances following non-trigger and trigger-positive assistant responses. Percentages are computed over repair-containing next turns within each condition.

occurred, not what form it took. We therefore next examine repair composition following trigger-positive and non-trigger responses. Because many repair turns are multi-label, we focus on whether key repair features occur anywhere within repair-containing next turns (Table 3). Repair following trigger-positive responses is more likely to include $R1$ -containing behavior, whereas repair following non-trigger responses more often includes $R3$ -containing behavior than repair following trigger-positive responses. This contrast is therefore not limited to single-label turns. One interpretation is that repetition/rephrasing ($R1$) reflects recovery-oriented interactional effort in response to visible assistant-side problems, whereas requests for more information ($R3$) remain more compatible with ordinary troubleshooting progression or other context-dependent follow-up behavior. The full distribution of repair-label combinations is provided in Appendix Table 11.

This context dependence is also visible at the interaction level. In one session about trace variables, the assistant remained largely topically relevant and non-trigger, yet the user repeatedly reformulated the same request across several turns. Although individual responses were partially helpful, the repeated reformulations and eventual negative feedback suggest task-level failure rather than successful task progression. This illustrates how repeated repair following superficially acceptable responses can reveal task-level failure not fully captured by isolated turn-level labels.

If repair is only weakly informative on its own, its diagnostic value may depend on the type of assistant response with which it co-occurs. We therefore next examine within-turn relationships by treating each paired interaction step $t = (U_t, A_t)$ as the unit of analysis. Here, repair is a property of the user utterance U_t , whereas trigger status and explicit feedback are properties of the correspond-

Repair	Non-trigger		Trigger	
	↓	↑	↓	↑
No Repair	6.5%	31.0%	8.5%	6.4%
Repair	4.9%	24.3%	16.7%	0.0%

Table 4: Explicit feedback rates on assistant responses, stratified by whether the corresponding user utterance U_t contains repair and whether the assistant response A_t is trigger-positive. Arrows denote thumbsDown (↓) and thumbsUp (↑). These rates are descriptive and should be interpreted cautiously, as some cells are based on small numbers of feedback events.

ing assistant response A_t . Table 4 summarizes this relationship by comparing feedback rates for repair versus no repair under trigger-positive and non-trigger responses.

Together, these results suggest that repair is not uniformly associated with explicit feedback. Instead, it becomes more diagnostically informative when it co-occurs with a trigger-positive response. Although based on relatively small numbers of feedback events, the strongest negative-feedback pattern appears when repair in U_t is paired with a trigger-positive assistant response A_t . A likely explanation is that repair in U_t reflects user effort to recover, while a trigger-positive A_t indicates that this effort was not successfully resolved, making explicit dissatisfaction more likely. Repair is therefore informative, but not sufficient on its own.

Dissatisfaction is concentrated in specific interactional configurations. Table 5 refines the within-turn negative-feedback pattern by providing a more fine-grained subtype view of repair in U_t while keeping the same paired-turn setup, with trigger status and feedback defined on A_t . The clearest pattern is for repetition/rephrasing ($R1$): negative feedback is highest when $R1$ co-occurs with a trigger-positive assistant response. $R2$ shows only a modest difference across trigger conditions, while $R3$ also increases under trigger-positive conditions but is harder to interpret robustly given sparse observations. This suggests that dissatisfaction, expressed by thumbsDown, is not associated with repair in general, but becomes more visible in specific repair-trigger configurations. A likely interpretation is that $R1$ reflects user attempts to recover from unresolved assistant-side deficiencies. When the paired assistant response is also trigger-positive, this may indicate that the user’s recovery effort was not successfully resolved, making explicit dissatis-

Repair Type	Non-trigger	Trigger
R1 (Repetition/Rephrase)	8.2%	18.9%
R2 (Info Add./Update)	7.1%	10.0%
R3 (Request More Info)	0.0%	14.3%

Table 5: Explicit negative feedback rates on assistant responses A_t by repair type in the user utterance U_t , stratified by trigger status of the assistant response. Repair subtypes lacking observations in one condition or with very low counts are omitted.

Model	Best Reg.	PR-AUC	SD
Repair-only	–	0.090	0.023
Trigger-only	–	0.109	0.021
Repair + Trigger	L_2 (0.01)	0.125	0.008
All Repairs + Trigger	L_1 (1.0)	0.127	0.034
All Repairs + Interaction	L_2 (0.01)	0.155	0.019

Table 6: Grouped session-level prediction of negative feedback using 3-fold cross-validation. Reported values show the best PR-AUC obtained for each feature set over a small regularization grid.

faction more likely.

Exploratory prediction provides additional support for the joint interpretation. We include a secondary exploratory analysis to assess whether repair and trigger signals carry usable predictive information for explicit feedback. This analysis is intended as a sanity check on signal usefulness, rather than as a predictive benchmark.

Repair alone provides limited signal for predicting explicit negative feedback. In grouped session-level evaluation (Table 6), repair-only models perform only slightly above baseline and below trigger-only models, while joint models combining repair and trigger information perform better. The strongest results are obtained when modeling repair subtypes together with repair–trigger interaction terms, mirroring the descriptive finding that repair becomes informative primarily in the presence of assistant-side deficiencies. Overall performance remains modest because explicit negative feedback is sparse. This is consistent with the interpretation that binary repair is too coarse to serve as a strong standalone signal, whereas subtype and context jointly recover more useful structure.

Repair is more selective for negative than for positive feedback. Explicit positive feedback (thumbsUp) is more frequent than negative feedback (21.2% vs. 7.8%; Table 7) and achieves higher overall predictability (PR-AUC 0.345 vs. 0.155). In contrast, predictive signal for thumbsDown is

Target	Rate	PR-AUC	Lift
thumbsDown	7.8%	0.155	+0.077
thumbsUp	21.2%	0.345	+0.133

Table 7: Comparison of explicit feedback targets. PR-AUC reports the best-performing model for each target; lift denotes improvement over the prevalence baseline.

concentrated in specific configurations, particularly repetition/rephrasing (*R1*) under trigger-positive conditions.

This asymmetry suggests that repair is more selective for negative outcomes: it becomes informative primarily in the presence of assistant-side deficiencies, whereas positive feedback reflects a broader, less interaction-specific signal.

Repair is mostly local, but non-local repair is substantial and varies with trigger type. To assess whether repair can be interpreted from local adjacency alone, we analyzed repair-target distance, defined as the number of assistant turns between a repair-containing user turn and the assistant turn it targets. Most repairs were local, targeting the immediately preceding assistant response (distance = 1, 104/157, 66.2%). However, a substantial minority were non-local: 25/157 (15.9%) had distance = 2, and 28/157 (17.8%) had distance ≥ 3 . Thus, roughly one third of repair-containing turns targeted an earlier assistant turn rather than the immediately preceding response, showing that local adjacency is informative but insufficient for inferring breakdown in dialogue.

This pattern holds after both non-trigger and trigger-positive assistant responses. Among repairs following non-trigger assistant turns, 33/107 (30.8%) targeted earlier context, compared with 20/50 (40.0%) among repairs following trigger-positive assistant turns. Non-local repair is therefore not confined to trigger-positive preceding responses, although it is somewhat more frequent after them.

Repair timing also varies descriptively by the trigger type of the preceding assistant response. Among repairs following trigger-positive assistant turns, context-loss cases were more often delayed than immediate (2 later vs. 1 immediate), whereas information failure was evenly split (8 vs. 8). By contrast, the remaining trigger types were more often followed by immediate repair. A complementary paired-turn analysis shows a related pattern: when the assistant response paired with a

repair-containing user turn is labeled as context loss, the repair is non-local in all observed cases (3 of 3). Together, these patterns suggest that context-level breakdowns may be especially associated with discourse-level repair, while other assistant-side deficiencies more often elicit turn-local repair.

Explicit negative feedback occurred in both immediate and earlier-context repairs (7 cases each). Because earlier-context repairs were less frequent overall, their observed thumbsDown rate was higher than that of immediate repairs (13.2% vs. 6.7%); however, this difference is not statistically significant in our sample (Fisher’s exact test: $p = 0.236$).

6 Discussion

Our findings show that repair is a context-dependent interactional signal rather than a direct indicator of failure. While predictive performance remains modest, the relative comparisons reinforce the descriptive findings: repair alone carries limited signal, whereas its usefulness emerges when combined with assistant-side context. From a dialogue perspective, repair can be understood as an update to conversational context rather than a breakdown event (Ginzburg, 2012), supporting an **interactional view of understanding**, in which alignment is established, negotiated, and repaired across turns rather than residing in individual model outputs. On this view, misunderstanding is not directly observable but must be inferred from interactional evidence such as user repair interpreted in context.

This helps explain why trigger annotations and explicit negative feedback are related but not interchangeable. The data show that most explicit negative feedback occurs on trigger-positive turns, but only a minority of trigger-positive turns receive such feedback, consistent with the sparsity and selectivity of explicit user feedback in deployed systems. A binary turn-level repair indicator is therefore a weak standalone proxy for dissatisfaction, whereas combining assistant-side deficiencies with finer-grained user response strategies yields a more informative signal, consistent with prior work on noisy and context-dependent interaction-derived feedback (Park et al., 2021; Pang et al., 2024). This is especially important because some annotated repair types, such as requests for more information, may remain compatible with successful ongoing task progression.

Repair-target annotations further show that a substantial minority of repairs address earlier conver-

sational context rather than only the immediately preceding turn, highlighting repair as a contextual rather than purely local signal. Our findings thus complement recent evidence that LLMs can lose track of context over the course of multi-turn interaction by showing how such difficulties become visible in user repair behavior (Laban et al., 2026). For deployed industrial assistants, this suggests that repair should not be treated as a direct failure indicator in isolation, nor should all repetitions be counted as equivalent signs of dissatisfaction. Instead, monitoring should combine assistant-side deficiencies with fine-grained user response patterns. In practice, configurations such as repetition/rephrasing following a deficient response or non-local repair in the presence of context-related trouble may be more useful candidates for review, escalation, or adaptive intervention than repair alone.

Implications for system design. Our findings suggest that deployed assistants should treat repair as a selective, context-dependent signal rather than a uniform indicator of failure. In monitoring, systems should prioritize patterns such as repetition/rephrasing following trigger-positive responses, while avoiding overinterpretation of all follow-up information requests as failures, since such requests may also reflect normal task continuation. Non-local repair may be especially useful for detecting breakdowns in context tracking or grounding and could motivate context reconstruction or targeted clarification. More generally, repair-aware response strategies could adapt to the type of user follow-up, for example by responding to repetition with clarification or reinterpretation rather than continued generation. Conversational assistants may also benefit from making user repair an explicit interaction affordance, allowing users to request clarification, explanation, or reinterpretation of assistant responses rather than relying only on reformulation. These implications align with prior work showing that conversational understanding is a two-way process and that implicit interaction signals can support improvement when interpreted carefully (Moore et al., 2024; Pang et al., 2024; Park et al., 2021; Stoyanchev et al., 2013).

7 Conclusion

We argued that user repair should not be treated as a direct proxy for misunderstanding in multi-turn LLM assistants. Using a human-annotated

dataset, we analyzed how assistant-side response deficiencies relate to subsequent user behavior and to sparse explicit feedback signals. Repair was more likely after deficient assistant responses, but it was also common after responses not annotated as deficient, where it often reflected ongoing task progression rather than breakdown. In particular, requests for more information were common after responses not annotated as deficient, reinforcing that not all repair-related follow-up behavior signals breakdown.

Repair alone was therefore not a reliable proxy for dissatisfaction or misunderstanding. At the same time, repair following responses not annotated as deficient suggests that interactional behavior may expose unresolved task-level difficulties less visible to response-level evaluation. User repair thus provides information not only about observable breakdowns, but also about possible limitations of turn-level assessment itself. The most informative signal emerged when repair was interpreted jointly with assistant-side deficiencies and, in particular, with selective repair forms such as repetition/rephrasing under trigger-positive conditions.

We also found that a substantial minority of repairs target earlier conversational context, highlighting the need for locality-aware monitoring. Overall, our findings suggest that breakdown and recovery in deployed human-AI dialogue are manifested indirectly through user repair in context, rather than through isolated repair behavior alone. Evaluating understanding in deployed assistants therefore requires joint analysis of assistant behavior, user repair, and conversational context.

8 Limitations

This study has several limitations. First, the analysis is based on interaction data from a specific industrial assistant setting, which may limit generalizability across domains, assistant designs, and user populations. Industrial language use is highly contextual, and repair patterns may differ in environments with other workflows, documentation structures, or expertise distributions. In addition, our repair annotation scheme intentionally includes some behaviorally defined follow-up categories, such as requests for more information, that are not always canonical repair in the strict conversation-analytic sense. This improves deployment relevance but also increases category heterogeneity.

Second, while repair behavior provides a useful observable interaction signal, it should not be interpreted as a direct or definitive indicator of root cause. Prior work on miscommunication similarly suggests that repair and other user follow-up behaviors can be informative while remaining ambiguous with respect to underlying causes (Purver et al., 2018; Sharma et al., 2026). A user correction, for example, may reflect retrieval failure, misunderstanding, missing context, or user-side reformulation preferences. More generally, repair has a dual role: it can signal assistant-side deficiencies but also function as a routine strategy during successful task refinement. Accordingly, interpretations linking repair behavior to assistant issues should be viewed as heuristic rather than causal.

Third, trigger labels reflect observable assistant-side deficiencies in interaction context, not validated system-level diagnoses. Categories such as *information failure* may therefore subsume multiple underlying mechanisms, including retrieval errors, grounding failures, missing context, or hallucination.

Fourth, many interactions are short, limiting visibility into longer recovery trajectories. The analysis is therefore primarily turn-centered rather than goal-centered. In practice, users may recover, abandon, or shift goals across multiple turns, requiring session-level analysis to capture outcomes more fully (Piskala et al., 2025).

Fifth, both repair behavior and explicit feedback provide only partial views of user experience. Users may abandon, switch channels, or silently tolerate poor responses without attempting repair or leaving negative feedback, meaning the observed signals capture only part of the breakdown space (Sharma et al., 2026).

Finally, the explicit negative-feedback analysis is based on a small number of instances, reducing statistical power and increasing variance across folds. Small performance differences should therefore be interpreted cautiously, and the results should be viewed as exploratory evidence about signal usefulness rather than deployable prediction performance. Future work should examine whether finer-grained distinctions among repair forms can more directly capture different classes of understanding failure, such as ambiguity, misreference, or context loss, and whether repair behavior varies systematically across users.

Ethics Statement

All annotators were compensated according to applicable national standards.

Use of AI Assistants. The lead author acknowledges the use of SiemensGPT, a Siemens internal AI assistant, for grammar correction, phrasing suggestions, and coding assistance.

Acknowledgements

Barbara Plank is partially supported by the European Research Council (ERC) Consolidator Grant DIALECT 101043235.

References

- Essam Alghamdi, Martin Halvey, and Emma Nicol. 2024. [System and user strategies to repair conversational breakdowns of spoken dialogue systems: A scoping review](#). In *Proceedings of the 6th ACM Conference on Conversational User Interfaces, CUI '24*, New York, NY, USA. Association for Computing Machinery.
- Herbert H. Clark and Edward F. Schaefer. 1989. [Contributing to discourse](#). *Cognitive Science*, 13(2):259–294.
- Jonathan Ginzburg. 2012. *The Interactive Stance*. Oxford University Press.
- Ulrich Gnewuch and Fabian Reinkemeier. 2026. [Overcoming breakdowns in customer-chatbot interaction: Design and impact of collaborative repair strategies](#). *Management Information Systems Quarterly*, 50(2):497–526.
- Ryuichiro Higashinaka, Masahiro Araki, Hiroshi Tsukahara, and Masahiro Mizukami. 2021. [Integrated taxonomy of errors in chat-oriented dialogue systems](#). In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 89–98, Singapore and Online. Association for Computational Linguistics.
- Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. 2015. [Fatal or not? finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2243–2248, Lisbon, Portugal. Association for Computational Linguistics.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2026. [Llms get lost in multi-turn conversation](#). In *International Conference on Learning Representations*, volume 2026, pages 54738–54778.

- Chengrui Li, Rujing Li, Yitong Bai, and Rui Li. 2026. [Towards multi-turn dialog systems for industrial asset operations and maintenance](#). *Preprint*, arXiv:2605.24953.
- Niharika Mathur, Tamara Zubatiy, Agata Rozga, and Elizabeth Mynatt. 2025. ["it feels like hard work trying to talk to it": Understanding older adults' experiences of encountering and repairing conversational breakdowns with ai systems](#). *Preprint*, arXiv:2510.06690.
- Biswesh Mohapatra, Seemab Hassan, Laurent Romary, and Justine Cassell. 2024. [Conversational grounding: Annotation and analysis of grounding acts and grounding units](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3967–3977, Torino, Italia. ELRA and ICCL.
- Robert J. Moore, Sungeun An, and Olivia H. Marrese. 2024. [Understanding is a two-way street: User-initiated repair on agent responses and hearing in conversational interfaces](#). *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1).
- A. Narimani and S. Klarmann. 2024. [Integration of large language models for real-time troubleshooting in industrial environments based on retrieval-augmented generation \(rag\)](#). In *Proceedings of the 7th European Industrial Engineering and Operations Management Conference*.
- Anh Ngo, Dirk Heylen, Nicolas Rollet, Catherine Pelachaud, and Chloé Clavel. 2024. [Exploration of human repair initiation in task-oriented dialogue: A linguistic feature-based approach](#). In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 603–609, Kyoto, Japan. Association for Computational Linguistics.
- Richard Yuanzhe Pang, Stephen Roller, Kyunghyun Cho, He He, and Jason Weston. 2024. [Leveraging implicit feedback from deployment data in dialogue](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 60–75, St. Julian's, Malta. Association for Computational Linguistics.
- Sunghyun Park, Han Li, Ameen Patel, Sidharth Mudgal, Sungjin Lee, Young-Bum Kim, Spyros Matsoukas, and Ruhi Sarikaya. 2021. [A scalable framework for learning from implicit user feedback to improve natural language understanding in large-scale conversational AI systems](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6054–6063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Rebecca Passonneau. 2006. [Measuring agreement on set-valued items \(MASI\) for semantic and pragmatic annotation](#). In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).
- Deepak Babu Piskala, Sharlene Chen, Udit Patel, Parul Kalra, and Rafael Castrillo. 2025. [Mind the goal: Data-efficient goal-oriented evaluation of conversational agents and chatbots using teacher models](#). *Preprint*, arXiv:2510.03696.
- Matthew Purver, Julian Hough, and Christine Howes. 2018. [Computational models of miscommunication phenomena](#). *Topics in Cognitive Science*, 10(2):425–451.
- Hossein A. Rahmani, Xi Wang, Yue Feng, Qiang Zhang, Emine Yilmaz, and Aldo Lipani. 2023. [A survey on asking clarification questions datasets in conversational systems](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2698–2716, Toronto, Canada. Association for Computational Linguistics.
- Kepa Joseba Rodríguez and David Schlangen. 2004. [Form, intonation and function of clarification requests in german task-oriented spoken dialogues](#).
- Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. [The preference for self-correction in the organization of repair in conversation](#). *Language*, 53(2):361–382.
- Abigail See and Christopher Manning. 2021. [Understanding and predicting user dissatisfaction in a neural generative chatbot](#). In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12, Singapore and Online. Association for Computational Linguistics.
- Omar Shaikh, Kristina Gligorić, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. 2024. [Grounding gaps in language model generations](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6279–6296, Mexico City, Mexico. Association for Computational Linguistics.
- Omar Shaikh, Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2025. [Navigating rifts in human-LLM grounding: Study and benchmark](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20832–20847, Vienna, Austria. Association for Computational Linguistics.
- Nikhil Sharma, Zheng Zhang, Daniel Lee, Namita Krishnan, Guang-Jie Ren, Ziang Xiao, and Yunyao Li. 2026. [Feedback by design: Understanding and overcoming user feedback barriers in conversational agents](#). In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26, New York, NY, USA. Association for Computing Machinery.

Joongmin Shin, Chanjun Park, Jeongbae Park, Jaehyung Seo, and Heuseok Lim. 2025. [MultiDocFusion : Hierarchical and multimodal chunking pipeline for enhanced RAG on long industrial documents](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20985–21004, Suzhou, China. Association for Computational Linguistics.

Svetlana Stoyanchev, Alex Liu, and Julia Hirschberg. 2013. [Modelling human clarification strategies](#). In *Proceedings of the SIGDIAL 2013 Conference*, pages 137–141, Metz, France. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. [Siren’s song in the AI ocean: A survey on hallucination in large language models](#). *Computational Linguistics*, 51(4):1373–1418.

A Annotation Guidelines and Examples

A.1 Trigger, User Response Strategy, Repair-Target Annotation

We annotate turns consisting of a paired user utterance and assistant response (U_t, A_t), along with prior conversational context when needed. The scheme captures observable user behavior following assistant output, focusing on user actions rather than inferred intent or system-internal causes.

Annotated fields. For each unit, we annotate:

- **Assistant response:** binary `is_trigger`
- **User turn:** one or more user response strategy (URS) labels
- **Repair target:** indicates the turn index of the prior assistant response addressed by the user turn; this field is used to derive repair-target locality (see below)

Assistant-side trigger labels. A response is labeled `is_trigger = TRUE` if it contains a user-relevant problem that could plausibly motivate repair, steering, or dissatisfaction (e.g., incorrect or irrelevant content, missing actionability, fallback responses, or loss of context). Otherwise, `FALSE`. Explicit feedback (`thumbsDown`) was available to annotators as context in ambiguous cases, but was not used as a deterministic labeling rule. Trigger annotations were based primarily on the observed quality and usefulness of the assistant response in context.

User response strategy labels. URS labels describe observable follow-up behavior and are organized into three functional families:

Repair.

- **R1 Repetition / Rephrasing:** Restates the same request without adding new information. Example: "What causes error F320?" → "Why does error F320 occur?"
- **R2 Information Addition / Update:** Adds or refines task-relevant details without explicit correction. Example: "It happens after 10 minutes."; "The machine is an S7-1500."
- **R3 Request for More Information:** Requests additional information related to the current task or topic. Example: "So how do I fix it?"; "What should I do next then?"; "Which part number exactly?"; "What is the current machine state?"
- **R4 Clarification Request:** Signals lack of understanding of meaning, reference, or reasoning. Example: "What do you mean by calibration drift?"; "Which parameter are you referring to?"
- **R5 Explicit Correction:** Indicates that the assistant’s response is wrong or based on a misunderstanding. Example: "No, the code is F320, not F302."; "That is incorrect; the log shows pressure loss."

The repair family is intentionally defined broadly as user actions that attempt to advance, restore, refine, or verify task understanding following an assistant response. While some categories such as repetition/rephrasing (*R1*) and explicit correction (*R5*) are more clearly associated with interaction breakdowns, others are inherently ambiguous. In particular, *R3* (Request for More Information) is defined behaviorally rather than evaluatively. Although annotators have access to the preceding assistant response, determining whether requested information is genuinely missing from the assistant’s answer is often difficult in technical troubleshooting dialogues, where information needs unfold incrementally across multiple turns and users frequently progress through diagnosis, explanation, and recovery steps.

Consequently, annotators label *R3* whenever a user requests additional task-relevant information, without requiring a judgment that the preceding response was objectively incomplete or deficient. Such requests may function either as repair following an insufficient response or as ordinary continuation within an ongoing troubleshooting process. We therefore treat *R3* as a repair-related behavior whose interpretation depends on assistant response quality and dialogue context.

Steering.

- ***S1 Output Restriction:*** Constrains format, scope, or verbosity of the response.
Example: "Just give me the steps."; "Keep it short."
- ***S2 Language / Terminology Control:*** Requests specific language or terminology.
Example: "Answer in German."; "Use HMI labels."

Task management.

- ***T1 Task Change:*** Initiates a new, independent task.
Example: "Never mind. How do I reset the PLC?"; "Forget that, what is the spare part number?"

Repair-target annotation. For repair-focused analyses, we additionally annotate `repair_target`, which records the turn index of the prior assistant response addressed by the user turn. For example, if, at turn 5, a user refers back to the assistant response from turn 1, then `repair_target = 1`.

This field enables us to distinguish between repair targeting the immediately preceding assistant response and repair referring to earlier interaction context. Accordingly, we derive *repair-target locality*: a repair is considered *immediate* (local) if `repair_target` corresponds to the immediately preceding assistant turn, and *earlier-context* (non-local) otherwise.

For *T1* (task change), `repair_target` is set to 0, indicating that the user initiates a new goal rather than addressing a prior assistant response.

Multi-labeling. Because user turns may serve multiple functions, multiple URS labels may be assigned, except for *T1*, which is single-label since it initiates a new task and cannot co-occur with repair behavior.

Setting	Metric	Value
Repair Target	Exact agreement	0.758
	Cohen's κ	0.701
Locality	Exact agreement	0.885
	Cohen's κ	0.754

Table 8: Inter-annotator agreement for repair-target annotation (`repair_target`) on repair-containing rows labeled by both annotators ($n = 95$). Exact agreement and Cohen's κ are computed treating target turns as categorical labels. Results are also shown for the coarse locality distinction used in the main analysis.

Core decision rules. Annotators distinguish: (i) new vs. ongoing task (*T1* vs. others), (ii) clarification request vs. request for information (*R4* vs. *R3*), (iii) correction vs. information update (*R5* vs. *R2*), and (iv) repetition vs. update (*R1* vs. *R2*). The guiding principle is to label observable user behavior rather than infer intent or system-internal causes.

For *R3*, annotators label the observable act of requesting additional information rather than determining whether the preceding assistant response omitted information necessary for task completion. When distinguishing *R3* from *R4*, annotators prioritize whether the user is requesting additional task-relevant content (*R3*) versus signaling lack of understanding of the assistant's wording, reference, or reasoning (*R4*).

For the first user turn in a session, no URS label is assigned because there is no prior assistant response to react to.

A.2 Agreement for Repair-Target Annotation

We assessed inter-annotator agreement for `repair_target` on repair-containing rows labeled by both annotators ($n = 95$). Because `repair_target` encodes the identity (turn index) of the referenced assistant response rather than a relative distance, agreement was computed using exact match and nominal Cohen's κ .

The results indicate substantial agreement on the exact target turn (exact agreement = 0.758, $\kappa = 0.701$), suggesting that annotators reliably identify the same conversational target underlying user repair behavior.

To assess agreement at a higher level of abstraction, we additionally derive a binary locality distinction: *immediate* (local), when the annotated target corresponds to the immediately preceding assistant turn, versus *earlier-context* (non-local),

when it refers to any earlier assistant response.

Overall, these results show that while annotators occasionally differ in selecting the exact target turn, they exhibit strong and consistent agreement on the broader locality of repair behavior.

A.3 Ambiguous Cases from Adjudication

Table 9 presents synthesized examples derived from proprietary annotation disagreements. We rewrote the utterances to preserve the ambiguity pattern while removing product-specific details. The cases illustrate recurring disagreements over whether a user turn should be labeled as task change ($T1$) or as a repair-oriented response strategy, whether multiple URS labels are simultaneously warranted, and whether the assistant response should be marked as a trigger.

B Descriptive Analyses

B.1 Repair Type Distribution

Table 11 shows the full distribution of repair-label combinations following non-trigger and trigger-positive assistant responses. Consistent with the main-text analysis, trigger-positive responses are more often followed by repetition/rephrasing ($R1$) and $R1$ -containing combinations, whereas non-trigger responses more frequently elicit requests for more information ($R3$). Multi-label combinations are relatively uncommon, and $R4$ and $R5$ occur only rarely. Overall, the distribution supports the view that repair behavior differs systematically by response context and that repair following non-trigger responses cannot be interpreted solely as recovery from observable assistant-side deficiencies.

B.2 User Behavior Following Negative Feedback

We analyze user behavior following explicit negative feedback (thumbsDown). As shown in Table 12, repair is the dominant response (71.4%), while task change remains a substantial alternative (28.6%); steering is rare (9.5%).

This indicates that even after signaling dissatisfaction, users typically continue the interaction by attempting repair rather than disengaging or switching tasks.

Trigger Type	R1	R2	R3	R4	R5
context_loss	0.25	0.25	0.00	0.00	0.50
entity_mismatch	0.00	1.00	0.00	0.00	0.00
info_failure	0.47	0.29	0.24	0.00	0.00
info_failure_plus_guidance	0.45	0.35	0.15	0.05	0.00
refusal_block	0.50	0.17	0.17	0.00	0.17
system_failure	0.67	0.00	0.33	0.00	0.00

Figure 3: Distribution of repair strategies in the next user turn, normalized within each trigger type. Each value represents the percentage of cases within that trigger type exhibiting the corresponding repair strategy.

B.3 User Behavior Following Trigger Responses

We next examine user responses to trigger-positive assistant outputs, treating them as implicit indicators of interaction breakdown. Focusing on the immediate next turn, we analyze both repair strategies and task-change behavior.

As shown in Fig. 3, the distribution of repair strategies differs across trigger types. The two information-related categories (information failure and information failure with guidance) show a mixture of repetition/rephrasing ($R1$), information addition/update ($R2$), and requests for more information ($R3$). By contrast, refusal and system-failure responses are more strongly associated with repetition/rephrasing ($R1$). These patterns should be interpreted cautiously for trigger types with few observations.

Task-change ($T1$) behavior also differs by trigger type (Fig. 4), with higher rates observed following information failures with guidance, suggesting that partially helpful responses may prompt users to refine or shift their task. Overall, these patterns indicate that user responses to trigger-positive outputs are structured and depend on the type of underlying breakdown.

C Exploratory prediction setup

We conduct an exploratory prediction analysis of explicit negative feedback using logistic regression models over binary features derived from the annotated repair and trigger labels. To assess how

ID	Ambiguity type	Synthesized user utterance	Synthesized assistant response	Why the case is ambiguous
A1	<i>T1</i> Task Change vs. <i>R3</i> Request for More Info	User: “How do I switch the system into production mode?”	Assistant: “To start production mode, first verify the machine is idle and all units are in their initial position. Then press <i>Start</i> .”	This turn can be read as <i>T1</i> because it starts a fresh procedural request, or as <i>R3</i> because it may ask for procedural continuation within the same troubleshooting thread.
A2	<i>R1</i> Repetition/Rephrase vs. no confident URS label	User: “Transition criteria?” User (later): “Criteria.”	Assistant: “No relevant information is available. Please provide more details.”	The later turn may be an abbreviated repetition (<i>R1</i>), but it is also so underspecified that some annotators may avoid assigning a confident URS label.
A3	Multi-label: <i>R1+R5</i> Repetition/Rephrase + Explicit Correction	User: “No, that is not correct. I need a replacement Z-axis for the line, not installation instructions.”	Assistant: “To install a new Z-axis, unlock the axes, move them manually, and store the new tool position.”	The user both corrects the previous answer and restates the need more clearly, supporting a joint reading as <i>R1</i> and <i>R5</i> . Annotators may still disagree on trigger status because the response is related, yet addresses the wrong subtask.
A4	Multi-label: <i>R1+R2</i> Repetition/Rephrase + Information Addition/Update	User: “Barcode reading error.”	Assistant: “A barcode reading error can have several causes. Please check code quality, scanner settings, lighting, and lens cleanliness.”	This turn can be read as a contextual rephrase of the earlier problem (<i>R1</i>) or as added specificity that narrows the issue to barcode reading (<i>R2</i>); both labels may be plausible. The assistant response is on-topic but generic, which can also create trigger disagreement.
A5	<i>R1</i> Repetition/Rephrase with trigger disagreement	User: “Is there currently any active error?” User (later): “Is there any active error right now?”	Assistant: “Yes, there is an active front-door error.”	The later user turn is naturally labeled as <i>R1</i> , but annotators may still disagree on whether the assistant response is a trigger because it is superficially responsive while potentially not satisfying the user’s underlying need for actionable status information.

Table 9: Synthesized examples of strong annotation disagreements, including multi-label URS cases.

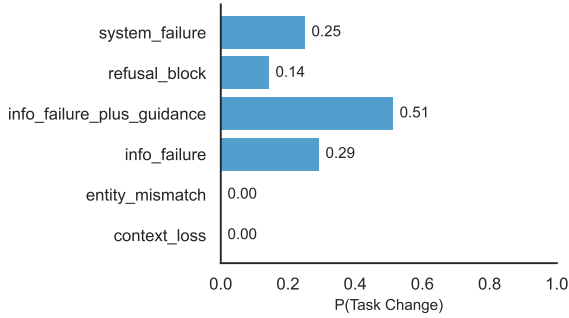


Figure 4: Task change rates by trigger type in the post-trigger setting.

much predictive signal is carried by different levels of interactional abstraction, we compare several feature families: repair-only baselines, trigger-only baselines, joint repair+trigger models, compact subtype-specific interaction models, and fuller models that include all annotated repair subtypes with and without subtype-specific repair-trigger interaction terms. All models use regularization; we sweep a small grid over L_1 and L_2 penalties with $C \in \{0.01, 0.1, 0.5, 1, 2, 5, 10\}$. We evalu-

ate all models with stratified grouped 3-fold cross-validation at the session level, ensuring that turns from the same session never appear in both training and test data. Because explicit negative feedback is rare, we treat PR-AUC as the primary metric and report ROC-AUC, F1, precision, and recall as secondary metrics. Reported values are means across folds. Given the small number of sessions containing explicit negative feedback, these experiments are intended to characterize the relative usefulness of different feature representations rather than to establish a high-confidence predictive benchmark.

Trigger Type	Description	Indicators / Heuristic
Information Failure	The assistant fails to provide a usable or relevant answer.	No relevant information; generic or misaligned response (e.g., definition instead of solution); hallucinated or unsupported content; ignores context. <i>Heuristic:</i> The user cannot proceed or complete the task.
Information Failure + Guidance	The assistant does not solve the task but provides partial information or recovery support.	Suggests next steps; requests clarification; provides partially useful but incomplete information. <i>Heuristic:</i> The user cannot complete the task but can make progress.
System Failure	The response is structurally broken or unusable.	Truncated output; broken formatting; malformed or incomplete tool responses.
Refusal	The assistant declines to answer without adequate justification.	Generic refusal without explanation; irrelevant constraint (e.g., unsupported language).
Entity Mismatch	The assistant refers to an incorrect or inconsistent entity compared to the user query.	Wrong machine, component, or identifier (e.g., user refers to PS42, assistant refers to PAS402); inconsistent terminology that changes the task context.
Context Loss	The assistant fails to incorporate relevant prior interaction context.	Ignores previously provided information; responds as if prior turns did not exist; reformulates the task without considering earlier inputs.

Annotation Heuristics. Annotate based on observable response behavior rather than assumed system capability. If a response does not help the user make progress, label it as *Information Failure*. If it provides partial guidance enabling progress, label it as *Information Failure + Guidance*. Use *Entity Mismatch* when the assistant introduces an incorrect task-relevant entity, and *Context Loss* when the assistant fails to incorporate previously established interaction context.

Table 10: Trigger annotation scheme with definitions and heuristics.

Label combination	Non-trigger	Trigger
[]	110	30
[R1]	33	23
[R3]	31	3
[R1, R2]	19	13
[R2, R3]	7	3
[R1, R3]	5	2
[R2]	4	2
[R5]	4	0
[R4]	2	1
[R1, R2, R3]	1	1
[R2, R5]	1	0
[R3, R5]	0	1
[R1, R2, R5]	0	1
Total	217	80

Table 11: Full distribution of next-turn repair label combinations following non-trigger and trigger-positive assistant responses. Empty brackets denote no repair label in the next-turn user utterance. Order-normalized combinations are merged (e.g., [R3, R1] is counted as [R1, R3]).

Behavior	Rate	Count
Repair	71.4%	15
Steering	9.5%	2
Task Change	28.6%	6

Table 12: Distribution of user behaviors following explicit negative feedback (thumbsDown).