A Prototype of Relational Risk Telemetry for AI-Based Tools and Chatbots

Radosław Kołacki
radoslaw@kolacki.eu

July 2026

Abstract

Conversational systems based on large language models are being deployed as the first line of contact in customer service, yet their quality is still largely measured using process metrics (response time, deflection rate, per-message sentiment) that do not capture phenomena with potential impact on relational outcomes: lack of dialogue progress, failed error-recovery attempts, or silent customer departure without complaint.

This paper presents a **prototype of a conceptual and implementational framework for conversational risk telemetry** (understood as the operationally observable dimension of broader relational risk), built on two components: an **Operational Taxonomy**—a taxonomy of conversational states and events designed so that every category is justifiable by textual evidence and maps onto an operational decision—and **LLM-as-a-Judge**—an automated measurement instrument operating on structured rubrics with an evidence-span requirement. The system measures **conversation dynamics** (how the conversation unfolded), not agent quality (whether the response was correct).

The prototype operates in a production environment and has processed a corpus of **1,146 sessions** (**8,374 messages**) from five main domains of the Polish market (plus 60 sessions from additional low-frequency domains). Empirical validation (inter-rater agreement, validation of associations with external outcome indicators, i.e., measurable business outcomes) is in progress; the validation plan is described in Section 7.

Methodological note: The results presented are preliminary observations from a pilot deployment. This paper does not constitute a report from a controlled validation study; its purpose is to present the concept of the tool, the prototype architecture, and directions for further validation.

Keywords: conversational AI, LLM-as-a-Judge, interaction risk, service recovery, emotion inference, relational telemetry, customer support automation

1. Introduction and Motivation

Conversations conducted by conversational AI systems have become part of the infrastructure of many services—from customer support and self-service to advanced business process assistance. In such an environment, "quality" of a conversation is not solely a matter of factual correctness of the response. Equally important is whether the user perceives progress, understands the next steps, and whether the interaction stabilizes the situation or, conversely, builds tension and distrust. From an organizational perspective, these are practical phenomena: they affect escalation of the situation, cost of service, repeat contacts, and ultimately purchase abandonment, high churn rates, or complaints.

Many popular approaches to measuring "emotion" in text—e.g., per-message sentiment, simple emotion classification, lexicon-based heuristics, and post-conversation surveys—can be useful, yet in the context of dialogue **may not always be sufficient** for operational decision-making. This is not because they are "bad," but because dialogue is a sequential phenomenon: the meaning of an utterance depends on what precedes it and on whether the conversation is actually moving the issue forward. Some signals that prove critical in practice are not about word choice in a single message but about dynamics: repetitions of intent, lack of progress despite successive turns, reactions to errors, sudden changes in communication style, or a "polite closing" that does not necessarily mean the problem has been resolved.

This paper grew out of project experience with a corpus of 1,146 conversational sessions (8,374 messages). In such material, one can observe recurring situations in which a single sentiment label or

simple emotion classification does not explain what matters most to a product or operations team: whether the conversation is heading toward resolution, whether escalation risk is growing, whether the user is losing trust, and at which point in the interaction a "flashpoint" emerges. The question, therefore, is not about the definition of emotion but about the fact that a useful diagnosis in practice requires tools that are sensitive to context and conversational trajectory.

For this reason, we adopt a narrower but more controllable perspective. We do not claim to be able to determine "what the user is really feeling" in a psychological sense. Instead, we are interested in what can be justified from the transcript: **interaction states and risks** that manifest naturally in language and dialogue structure and that can be linked to operational decisions (e.g., when to escalate to a human, how to design recovery responses, how to monitor quality degradation). In the following sections, we describe a set of categories for such states and hypotheses about the signals by which they can be predicted. We then describe how to scale measurement and annotation using an **LLM-as-a-Judge** approach based on rubrics and structured justifications, so that the output is not only numerical but also auditable.

1.1. The Practical Problem

In customer service, emotion is not "background" to the conversation but one of the main mechanisms determining whether the interaction ends in relationship repair or conflict escalation. The scale of the phenomenon is empirically well documented: in a U.S. study (National Customer Rage Study, 2025 edition) **77%** of respondents reported experiencing a product or service problem in the past 12 months, and **64%** reported feeling "rage" as a result; **50%** admitted to raising their voice, and digital channels have become the dominant medium for complaints (**45%** vs. **33%** phone), with a significant "public" escalation vector (social media posts) and reported lack of firm response (**43%** of cases) (Broetzmann and Beinhacker, 2025). These data pertain to the U.S. market and should not be directly transferred to the Polish context; we cite them as an illustration of the scale of the phenomenon. They point to two characteristics of the environment in which conversational systems operate today:

1. a high level of readiness to escalate, and
2. a shift of conflict to text-based channels, where micro-signals of emotion must be read without the support of body language and prosody.

This context intersects with the practice of service automation: chatbots (including LLM-based ones) are deployed as cost-scalable "first lines" of contact, but regulator reports show that their effectiveness declines with problem complexity, and poorly designed pathways can trap customers in loops of repetitive, unhelpful responses ("doom loops") without a meaningful exit to a human (Consumer Financial Protection Bureau, 2023). From the customer experience perspective, this is not a neutral usability defect: it is a situation in which the service tool becomes part of the problem and raises the psychological cost of contact.

In the service recovery literature, there is a precise term describing this mechanism: **double deviation,** i.e., a second failure following the original one, when the firm cannot adequately repair the error and the repair attempt itself becomes another failure (Smith and Bolton, 1998). Crucially, the "second failure" does not operate additively. Research indicates that double deviation **intensifies the trust violation** caused by the first failure and requires different recovery strategies than a single "deviation" (in particular, greater weight on apologies and non-recurrence promises vs. financial compensation alone) (Smith and Bolton, 1998). If a conversational system is to serve as the "front office," then its errors—especially during escalation—should be treated not as ordinary intent-classification errors but as a potential generator of a second failure within the logic of the relationship. At the same time, the emotion we wish to measure does not reduce to a single "positive–negative" dimension. In research on post-sales behaviors, the distinction between "dissatisfaction" and the more specific emotion of **anger** and its link to retaliatory behaviors ("getting back") in services is emphasized (Blodgett et al., 1997). This distinction is practically relevant: in customer service, we are interested not only in *whether* a conversation is negative but *whether* it enters an escalation trajectory where the **likelihood** of actions costly to the brand (churn, public complaints, retaliation, etc.) increases.

At this point, the working thesis of the paper emerges, formulated cautiously—not as a psychological resolution but as a hypothesis based on deployment observations and conversation analysis: **classical approaches to "emotion in text" (e.g., simple sentiment, emotion lexicons, basic categories) may not be sufficient to capture escalation transition moments or to assess the quality of recovery actions in dialogue.** Consequently, we need an operationalization of emotion that (a) is useful in text-based channels, (b) accounts for conversation dynamics, and (c) can be measured in a comparable

and scalable manner. The tool realizing this operationalization is **LLM-as-a-Judge** as an evaluative layer: strong models can be used as "judges" to assess response quality and compliance with criteria (e.g., de-escalation, empathy, adequacy of repair). In comparative benchmarks analyzed by Zheng et al. (2023), GPT-4 achieved over 80% agreement with human preferences. In parallel, synthetic treatments of this practice and its limitations exist, providing the approach with a solid analytical framework (Liu et al., 2023).

1.2. Emotion

The word "emotion" in this paper is a **shorthand** and does not aspire to be a psychological diagnosis. We are interested not so much in the hypothetical internal state of the user as in a **measurable interaction state**—one that can be justified by evidence in the text and the course of the conversation. From the perspective of the theory of constructed emotion, we move away from searching for universal "fingerprints" hidden in the brain or body, because research shows that variability is the norm (Barrett, 2017). Instead, we focus on **operational states**: mounting tension, loss of a sense of agency, transition to escalation mode, withdrawal (including "silent churn"), as well as de-escalation and return to cooperation. This approach helps avoid an error that in this paper—inspired by the theory of constructed emotion (Barrett, 2017)—we tentatively call the **mental inference fallacy** (the term itself was introduced for the purposes of this paper). This principle forms the foundation of Evidence-Only Judging, described in detail in Section 4.1.

1.3. Unit of Analysis: Turn / Conversation Window / Full Session

Analyzing the service process requires monitoring the dynamics of change at three temporal levels, enabling us to capture emotion as a "movie" rather than a "snapshot":

1. **Turn (Micro-scale):** Analysis of a single message for textual escalation signals and local affect indicators, identified based on lexical and pragmatic features (Sanguinetti et al., 2020) and sequential patterns (Sandbank et al., 2018).
2. **Conversation Window (Meso-scale):** Context of the last few turns, necessary to detect the **No-Progress Loop** phenomenon, in which the system repeats ineffective responses while ignoring the user's growing confusion.
3. **Full Session (Macro-scale):** A cumulative perspective enabling measurement of cumulative interaction quality (Smith and Bolton, 1998). At this level, we assess the final interaction effect (`final_state`) compared to the initial one (`initial_state`), which allows detection of **Double Deviation** (Smith and Bolton, 1998, see ch. 1).

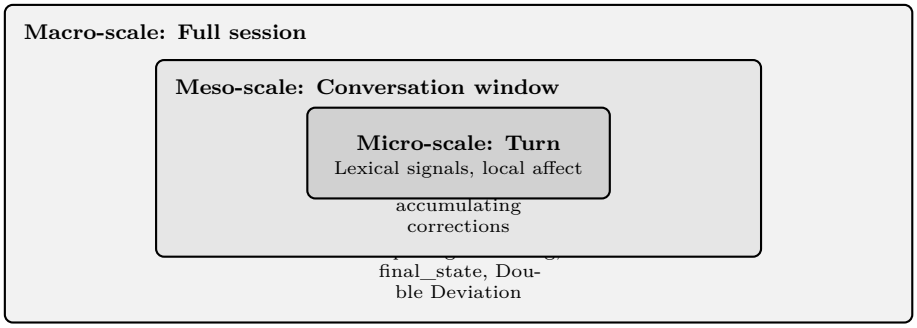

Figure 1: Three temporal scales of conversation analysis. The turn (micro) provides local signals; the conversation window (meso) enables detection of loops and mounting friction; the full session (macro) allows assessment of the cumulative relational outcome.

1.4. Assumptions and Limits of Generalization (Domain, Channel, Language, Context)

Although the mechanisms of emotion construction are part of human biology, their interpretation is closely dependent on social and domain context (Barrett, 2017).

- **Domain and Context:** Customer reactions are embedded in their expectations of a given service—what is perceived as fair in a hotel may differ from expectations in e-commerce (Blodgett et al., 1997).
- **Channel:** Text lacks acoustic parameters (e.g., pitch) that objectively signal stress; therefore, models based solely on text must account for this gap and should not claim accuracy comparable to multimodal analysis.
- **Language and Culture:** Emotions are not universal constructs—Hofstede et al. (2010) demonstrate systematic cross-cultural differences in emotional expression, politeness norms, and confrontation styles. This has direct operational consequences: what constitutes a direct complaint in one culture may look like a neutral information request in another. Without accounting for local communication norms, a system risks systematic misclassifications that lead to under-prioritization of sessions requiring urgent intervention (Barrett, 2017; Hofstede et al., 2010).
- **Scope Limitations:** This paper focuses on the text channel, the Polish market, and five main domains that constitute the corpus source (plus 60 sessions from additional domains). Extension to other languages, domains, and channels requires separate calibration of rubrics and threshold parameters. We also assume that the AI judge evaluates interaction according to rigorous quality rubrics rather than "sensing" feelings—which is a fundamental condition for the system's robustness against biases built into language models.

2. Empirical Material and Context

In real-world deployments of **conversational AI**, we do not work with laboratory data or *one-shot classification* tasks but with transcripts of interactions embedded in a business process, with its constraints (policies, SLAs, GDPR, limited escalation paths, changing sources of truth, etc.). For this reason, a rigorous description of the data and the context of their creation becomes part of the methodology, not an editorial supplement.

In the ML/NLP literature, there is growing emphasis on standardizing data documentation and conditions of use: both in the form of **"datasheets"** (motivation, composition, collection process, applications, and risks) and **"data statements"** (characteristics of the population, language, channel, and text production context), precisely to limit over-interpretation and false generalizations. In this spirit, we treat our material as an **observational corpus**: its value lies in revealing typical escalation and repair mechanisms under production conditions, while simultaneously requiring caution in drawing inferences beyond the domain, channel, and language.

2.1. Corpus Characteristics: Transcripts and the Dynamics of the Polish Service Market

The starting point for building the system is authentic material from production conditions, not laboratory data—which is critically important for the accuracy of rubrics and threshold calibration. The research material consists of a corpus of **8,374 messages** grouped into **1,146 conversational sessions**, originating from customers in Poland across industries such as healthcare, sports (fitness coaches), mobile application help centers, and B2B (business-to-business) sales chats on websites. After applying quality filters, **1,146 sessions** were qualified for detailed analysis—this number is presented in Table 1. Inclusion criteria: (1) the session contains a minimum of two messages (at least one user turn and one assistant turn); (2) the session is flagged as `mvp_v2_complete`, meaning the scoring pipeline processed the session without errors and returned a complete JSON output with all schema fields (Table 7). Sessions rejected due to incomplete output (e.g., API timeout, JSON parsing error) are excluded from the analytical set. The full filtering workflow is as follows: (1) initial pool: all sessions recorded in the system; (2) rejection of single-message sessions (fewer than two messages); (3) rejection of sessions with erroneous or incomplete scoring output (`scoring_failed` or missing JSON fields; $n = 0$ in the current set); result: **1,146 analytical sessions**. The number of single-message sessions rejected in step (2) is not recorded in the current pipeline version and will be added in the next iteration.

An important methodological assumption is that **sessions consisting of only one message are excluded from analysis**, because every conversation is initiated by the AI Assistant. In this model, a single-message session indicates only that the bot started an interaction without any response from the customer, which does not allow for analysis of a relational trajectory. Such sessions may, however, constitute a separate class of events related to interaction abandonment and should be analyzed separately

Metric	**Medical**	**Sport/Fit.**	**IT**	**Educ.**	**Travel**	**TOTAL**
Sessions	186	437	232	125	106	**1,146**
Median msgs/session	5.0	5.0	3.0	4.0	5.0	**5.0**
Mean msgs/session	6.3	8.4	6.3	6.7	7.4	**7.3**
P90 msgs/session	12	17	13	12	12	**13**
% effort = 3	90.9%	92.2%	85.3%	96.0%	77.4%	**86.8%**
% with loop	1.1%	0.5%	0.4%	0.0%	0.0%	**0.4%**
% with exit intent	0.0%	7.6%	1.7%	0.0%	0.0%	**3.2%**
P0/P1/P2/P3 dist.	0/12/4/84%	8/7/5/80%	1/18/4/77%	0/13/6/81%	0/2/6/92%	**3/10/5/82%**

Table 1: Descriptive statistics of the corpus (updated: July 15, 2026). Data from the production system (n = 1,146 sessions). Effort scale: 1–5 (values 4 and 5 did not occur in the current dataset). Columns: Medical (186 sessions), Sport/Fitness (437), IT (232), Educational (125), Travel (106). The total also includes the remaining domains (60 sessions).

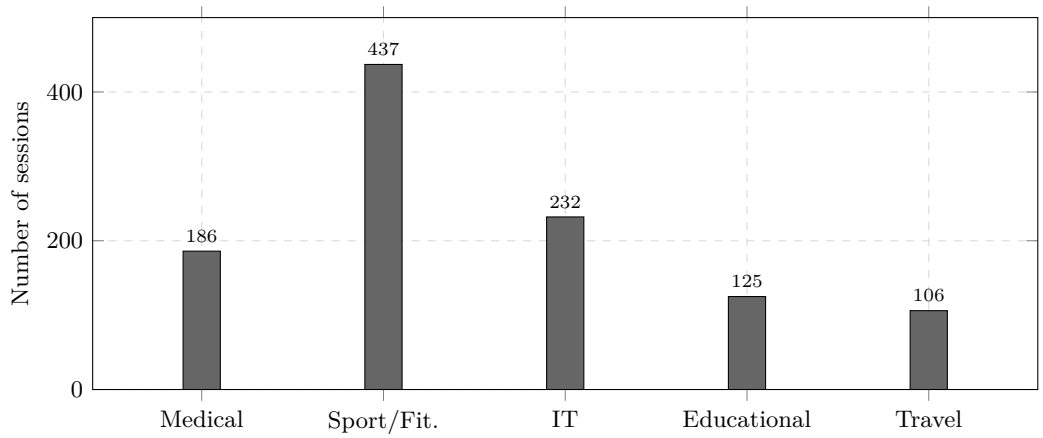

Figure 2: Distribution of sessions in the corpus by domain (n = 1,146). The Sport/Fitness domain constitutes the largest segment (38.1%), reflecting the specifics of the client base. The sum of the five named domains is 1,086; the remaining 60 sessions come from low-frequency domains.

in future iterations. Consequently, the actual AI judge analysis begins with the **second message in the session**—the customer's first response.

During the taxonomy design phase, prior to launching the current version of the automated pipeline, a qualitative review of conversations revealed several recurring patterns. The most commonly identified event was the **No-Progress Loop**, identified across most domains, though with different linguistic markers: in the medical sector, it manifested primarily through neutral reformulations of the same procedural question, while in e-commerce through noticeably shortened messages and increasing directness of demands. The **Justice Gap** pattern was most visible in sessions related to account and subscription management, particularly when the system refused without explaining the procedure—which corresponds to the findings of Blodgett, Hill, and Tax on the priority of procedural over distributive justice (Blodgett et al., 1997). The **Failed Recovery** pattern was observed in the technical support domain, where the system repeatedly suggested diagnostic steps already performed by the user. *Note:* the above observations come from a manual qualitative analysis conducted during the rubric design phase, not from the automated judge's output. In the current pipeline, the `failed_recovery` field exhibits zero variance (Section 7.1), which suggests a problem with automated detection, not absence of the phenomenon. These observations are inductive in nature—they influenced the shape of the taxonomy and the threshold selection described in Section 4 but are not statistically representative and require formal validation. The focus on the user's first response is important for several reasons. The architecture described in this paper operates in **offline** mode (post-hoc transcript scoring), designed with the possibility of extending to real-time mode in a subsequent iteration.

- **Prevention of "Silent Churn":** Failure to de-escalate at the first step often results in permanent customer withdrawal from the relationship—without any explicit complaint that could trigger intervention (Technical Assistance Research Programs [TARP], 1986). (The TARP findings date from 1986; extending the "silent churn" mechanism to contemporary digital channels, where switching barriers are lower, is a hypothesis requiring separate validation.) This phenomenon is particularly costly from a business perspective because it leaves no signal in classical complaint-based metrics.
- **Early Detection of Relational Risk:** A design hypothesis inspired by the theory of Barrett (2017) posits that in industries such as healthcare, uncertainty and lack of clear explanations may be stronger triggers of negative affect than wait time alone. Different domains therefore require different threshold parameters for the same taxonomic categories.
- **Avoiding Double Deviation:** Post-hoc assessment of the first error-repair steps enables detection of patterns in which a failed recovery attempt may intensify the negative assessment of the original failure (Smith and Bolton, 1998). Smith and Bolton (1998) showed that the manner of recovery execution affects the assessment of the entire experience, and a disappointed expectation of correction may be judged more harshly than the original error itself.

Based on observations from such diverse material—from technical support to administrative operations of medical facilities—the judge's rubrics were designed to account for the domain-specific nature of signals: the same emotional category (e.g., anger) in different industries constitutes a unique "instance" rather than a fixed fingerprint (Barrett, 2017). The judge interprets a given category in a domain context based on instructions contained in the rubric, not through fine-tuning on the corpus. This architecture is intended to support the generation of structured **relational telemetry** from the very first customer contact with the system, which may enable earlier prioritization of sessions potentially associated with churn risk (TARP, 1986).

2.2. Data Preparation and Annotation Protocol: From Transcript to Measurement Material

If emotions are to be measured in a comparable manner, the conversation transcript must be treated as data with a defined structure rather than a "description of a situation." In practice, this means that even before the actual evaluation, we need two layers of order:

- first, a **documentation layer** that clearly describes where the data came from and what they may be used for;
- second, an **operational layer** that transforms the conversation into units of observation (turns/windows/sessions) and into a set of variables that can later be aggregated and tested.

In the documentation component, we draw on practices developed in ML/NLP: "datasheets for datasets" as a standard for describing motivation, composition, collection, and risks of a dataset, and "data statements" as a form of disclosing the linguistic and population conditions on which generalizability depends. These frameworks are a methodological tool: without them, it is very easy to attribute universal emotion

signal status to a user's linguistic behavior, when in reality it is a product of the channel, domain, time pressure, or style norms. In a parallel manner, at the level of evaluative models, the logic of "model cards" is useful: explicit specification of purpose, limitations, and test conditions rather than the tacit assumption that the measure "always works." The operational layer begins with standardizing transcripts to a common format: roles (user/system), turn order, basic metadata (channel, case type if available), and timestamps. Only on such material can one build "observations" that are not essay-style quality descriptions but **completions of structured quality-assessment rubrics that map interaction dynamics onto specific numerical values and logical flags**. A key element of the annotation protocol is the deconstruction of each session through the lens of three justice dimensions:

- **distributive** (whether the outcome is perceived as fair),
- **procedural** (whether the process was efficient and transparent),
- **interactional** (whether the system treated the customer with empathy and respect), enabling prediction of future customer behaviors (Blodgett et al., 1997; Tax et al., 1998).

This protocol rigorously enforces the Evidence-Only Judging principle (see Section 4.1), requiring the AI judge to identify operational states solely on the basis of evidence present in the text (Barrett, 2017). The final product of this process is a **JSON data structure** containing an automated **Priority Score (PS)** indicator, which in the MVP version prioritizes escalation of a given case solely on the basis of conversational signals (future expansion to include business metadata such as customer lifetime value CLV or SLA priorities is planned—see Section 3.3). This structural preparation provides the foundation for precise AI judge analysis from the user's very first response in the session.

2.3. Corpus Representativeness: Selection Biases and Diversity of Interaction Types

The most important methodological risk in working with service data stems from the fact that **conversation transcripts do not constitute a representative sample from the general user population** but are the result of a strongly non-random selection process (Gebru et al., 2021). Transcripts primarily cover situations in which users engage in interaction in connection with a difficulty, ambiguity, or need that they cannot fulfill on their own (TARP, 1986; Broetzmann and Beinhacker, 2025). It should be emphasized, however, that the **selection mechanism does not restrict the corpus exclusively to interactions with extreme emotional intensity**. Analysis of 8,374 messages allows identification of several types of interactions, each carrying different risk for the relationship with the user:

- **Service/Product Inquiries:** Questions such as "how much does the product cost?" or "how do I choose a plan?" are not merely requests for data. In the predictive processing framework (Clark, 2016), any ambiguity in the response may generate a prediction error that burdens the user's cognitive resources.
- **Transactional/Administrative (Account and Subscription Management):** Requests to cancel a subscription, obtain a refund, or change a plan are critical tests of **procedural justice** (Blodgett et al., 1997; Tax et al., 1998). If the system impedes these processes or fails to recognize intent, customer affect centers on a sense of procedural injustice, which in post-complaint behavior research has been significantly associated with repurchase intention and negative *word-of-mouth* (Blodgett et al., 1997; Tax et al., 1998; TARP, 1986).
- **Usability/Navigation Support:** Questions like "where can I find my completed workouts?" indicate friction in achieving user goals. Lack of immediate informational gratification can cause mounting negative valence. The AI judge must detect the moment when a minor inconvenience turns into operational frustration—a transition often invisible in classical sentiment analysis because the user's language remains neutral.
- **Tech Support/Service Failure:** Application bug reports are classic **Service Failure** situations (Smith and Bolton, 1998). Research on *service recovery* suggests that the manner of an organization's response to a reported error can have significant impact on subsequent experience evaluation and declared loyalty (Smith and Bolton, 1998). A failed recovery attempt leads to the **Double Deviation** phenomenon, potentially intensifying the negative evaluation and affect caused by the original failure (Smith and Bolton, 1998; Bougie et al., 2003). In this niche, the risk of retaliatory behaviors, including negative WOM, may also scale (Bougie et al., 2003).
- **Personal Support (Motivational and Affective Needs):** Messages such as "I'm sleepy, I need motivation!" can be interpreted—in the framework of the theory of constructed emotion—as a report of an **affective state** (low valence, low arousal) and depleted body budget of the user (Barrett, 2017).

In this niche, the role of the AI system shifts from pure information toward active emotional regulation through appropriate communicative signals.

In summary, although the corpus is "filtered" by the need for contact, it contains a diverse cross-section of intents present in the data. The judge's rubrics were designed to account for this diversity: "subscription cancellation" may be a cool and technical process, while "a question about 4 eggs in a diet" may carry a significant load of uncertainty about one's own health goals.

3. Related Work

The approach proposed in this paper is situated at the intersection of three mature yet dynamically evolving research streams, each contributing important methodological tools while simultaneously possessing specific limitations in the context of operational customer service. The first is **classical sentiment and opinion analysis**, which for decades has focused primarily on binary or simple positive–negative polarity scales. Traditional methods, based on lexicon models or algorithms such as SVM (Support Vector Machine), typically operate at the level of a single document or sentence, which in service contexts precludes understanding the evolution of customer affect in response to successive system messages. Although contemporary affect research emphasizes that valence (pleasure/displeasure) accompanies humans throughout life, determining sentiment alone is merely a "partial picture" of the emotional state. In the service context, this approach often overlooks the process of fully reconstructing dialogue dynamics, precluding effective detection of phenomena such as **Double Deviation** (see Section 1).

The second stream is **emotion analysis in NLP**, a field growing extremely rapidly but still lacking a stable methodological consensus regarding the scope of applied theoretical frameworks. Researchers continue to oscillate between classical models of discrete basic emotions (Ekman) and more complex systems such as Plutchik's wheel of emotions, leading to terminological fragmentation. A key challenge remains incorporating **cultural and demographic context;** as the theory of construction indicates, without rigorous separation of text form from hypothetical internal state, these systems are vulnerable to the **mental inference fallacy** error (Barrett, 2017). It is worth noting that this argument is self-referential: if perceiving emotions in others is an act of categorization rather than detection, then the LLM judge—evaluating a transcript—also constructs emotions based on its training data and prompt rather than "reading" them from the text. This self-referentiality provides additional justification for the Evidence-Only Judging principle described in Section 4. Furthermore, standard NLP approaches rarely account for the phenomenon of **emotional granularity**, i.e., the ability to construct precise and differentiated states (e.g., distinguishing dejection from irritation), which directly affects the accuracy of system responses in crisis situations (Barrett, 2017).

The third stream, constituting the direct context for the AI judge described in this paper, concerns **evaluation of dialogue systems using strong language models (LLM-as-a-Judge)**. Under benchmark conditions, high agreement of such models with human preferences is reported, making them an attractive tool for automated conversation quality assessment. Nevertheless, the literature increasingly describes typical sources of errors and biases in such assessments, arising in part from the fact that these models inherit demographic and linguistic biases from their training data, which can lead to systematic unfairness, e.g., erroneously evaluating a specific communication style as aggressive. The proposed approach integrates these three streams within a relational telemetry system: the operational taxonomy provides risk categories, LLM-as-a-Judge provides a scalable measurement instrument, and a set of KPIs transforms measurement results into operational decisions. In this way, the AI judge becomes a link connecting text analysis with the operational need to protect customer loyalty and prevent **silent churn**.

3.1. Sentiment and Emotion Analysis in NLP: Capabilities and Limitations

In the **sentiment/opinion mining** tradition, "emotion" is most often reduced to a simple dimension of affective attitude: positive or negative polarity, possibly supplemented by intensity. Such an approach is pragmatic and works well in low-dynamics tasks such as aggregating opinions in product reviews, but exhibits critical limitations in the analysis of service conversations (Smith and Bolton, 1998). In interactions with conversational systems, **the tone of an utterance is often secondary to the lack of progress in the dialogue**; a user may maintain linguistic neutrality while mounting lack of progress builds frustration beyond the text level. It is worth noting that sequential approaches exist (including LSTM and BERT models applied to turn sequences) that partially address the dynamics problem; however,

their limitation remains a focus on emotion or sentiment classification *per utterance* rather than assessing repair quality across the entire session. A separate stream consists of automatic dialogue evaluation metrics such as USR (*Unsupervised and Reference-free*) and FED (*Fine-grained Evaluation of Dialogue*), which evaluate conversation quality without requiring reference responses (Mehri and Eskenazi, 2020a,b). In parallel, the DBDC (*Dialogue Breakdown Detection Challenge*) series proposed formal frameworks for detecting moments when dialogue ceases to function (Higashinaka et al., 2016). These approaches are closer to the proposed telemetry than classical sentiment but focus on generation quality (whether the response is coherent, relevant, interesting) rather than relationship dynamics (whether the interaction repairs the problem and protects trust)—which represents a different analytical perspective. Moreover, the escalation process need not manifest as constant "negative sentiment" in every turn—mounting frustration is often invisible in individual messages and only reveals itself in sequence dynamics. The literature indicates that it is not the original *service failure* but the bungled recovery attempt—**Double Deviation** (Smith and Bolton, 1998)—that generates the most destructive affective states.

The growing field of **emotion analysis** in NLP goes a step further, offering classifications of specific states (e.g., anger, fear, sadness) but simultaneously struggling with a lack of unified terminology and stable consensus on theoretical frameworks (Buechel and Hahn, 2016). Recent reviews of the field emphasize ambiguity regarding what exactly is being measured: the author's mental state, the emotion evoked in the recipient, or emotion as an objective property of the text (Barrett, 2017). From the perspective of the theory of construction, every emotional category (e.g., "Anger") is in reality a population of highly diverse instances, meaning that anger at an error in a mobile app will have entirely different linguistic markers than anger at lack of a doctor's appointment (Barrett, 2017). The lack of rigorous separation of text form from hypothetical internal state means that NLP models often fall into the trap of the **mental inference fallacy** (Barrett, 2017, see Section 1). Consequently, even when a model "recognizes emotions," the result remains difficult to compare across domains: what is considered motivational arousal in the fitness sector may be classified as aggressive impatience in the medical sector (Barrett, 2017; Hofstede et al., 2010).

The proposed relational telemetry aims to fill this gap by moving beyond a catalog of labels toward analysis of **procedural and interactional justice** (Blodgett et al., 1997; Tax et al., 1998). The system must understand that the user's goal is not so much to express feelings as to accomplish a specific task, and every ambiguity in the system's response generates cognitive cost and trust erosion—which the AI judge must register before potential loyalty loss occurs.

3.2. Conversational Signals: Dialogue Dynamics, Escalations, and Error Recovery

If in this paper we treat "emotions" as a measurable interaction state, the natural anchor points are not declarations such as "I'm angry" but signals that the conversation has stopped working: the user's cognitive cost is increasing, task progress is declining, and the system cannot transition from error to repair. In the **dialogue breakdown** literature, such a situation is sometimes defined as the moment when a participant can no longer meaningfully continue the conversation in the given direction (Higashinaka et al., 2016). Assessing whether a dialogue "breakdown" has occurred requires measures based on observable sequential markers—not on inferences about mental states—to avoid the **mental inference fallacy** (see Section 1). In analytical practice, it is convenient to operationalize these phenomena as conversational signals at three levels:

1. **Dialogue dynamics (whether the conversation "moves forward").** The most empirically stable signals are sequential in nature: user-side repetitions and reformulations, "ping-pong" of the same intents, and increasing turn length without problem resolution. Each such turn generates cumulating cognitive cost (Clark, 2016). In production data, this "accumulation of corrections" (when the user clarifies and corrects the system) often precedes frustration even when the text itself contains no explicitly negative words, causing classical sentiment models to "miss the problem" (Smith and Bolton, 1998).
2. **Escalations.** Escalation is a critical signal in which the user transitions from attempts to repair the dialogue to an explicit demand for an alternative channel or connection to a human. A failed escalation at the first step may result in **Silent Churn**, potentially weakening customer loyalty (TARP, 1986). The AI judge uses these signals to compute the **Priority Score (PS)** indicator, which synthesizes detected affect with the business risk of violating interactional and procedural justice principles (Blodgett et al., 1997; Tax et al., 1998).
3. **"Failure recovery" (whether the system can recover from error).** The ability to recover from error (*Service Recovery*) is a key moment of truth in the relationship (Smith and Bolton, 1998). If the

bot cannot effectively respond to a reported problem (e.g., a technical error or unclear offer), **Double Deviation** occurs (Smith and Bolton, 1998). On the other hand, a smoothly executed recovery may in some cases lead to the phenomenon known as the **Service Recovery Paradox**—a situation where the quality of crisis management results in higher declared loyalty than before the error occurred. This phenomenon is, however, empirically controversial and does not occur universally (Smith and Bolton, 1998). The AI judge monitors this process by measuring whether the system provides appropriate explanations and empathy, which in the service recovery literature is indicated as an important element of effective relationship repair (Smith and Bolton, 1998).

3.3. LLM-as-a-Judge: Scalable Conversation Evaluation and Bias Control

The contemporary turn toward **LLM-as-a-Judge** methodology arises from the need to find a scalable and cost-effective alternative to human evaluations in the evaluation of generative systems. In comparative benchmarks analyzed by Zheng et al. (2023), GPT-4 achieved over 80% agreement with human preferences, making language models a promising tool for analyzing large conversational datasets. However, it should be remembered that these models are not neutral observers and exhibit specific cognitive errors, such as **position bias** (privileging responses at a specific position in the prompt) and **verbosity bias** (a tendency to rate longer and more "verbose" utterances higher). An additional challenge is the **self-enhancement bias** phenomenon, in which the model tends to favor texts with structure and style similar to its own generations. To mitigate these risks, Liu et al. (2023) propose in the **G-Eval** methodology the use of structured **evaluation rubrics** and explicit **Chain-of-Thought** reasoning steps. This approach forces the judge model to decompose the customer's complex affect into smaller, measurable components and to base its verdict on specific **textual evidence** (text spans). Evaluations can take the form of **pointwise** (individual assessment), **pairwise** (comparing pairs of utterances), or **listwise** (ranking entire lists), with each configuration requiring rigorous calibration to avoid model overconfidence (Liu et al., 2023). In parallel, specialized judge models are being developed (e.g., JudgeLM (Zhu et al., 2023), Auto-J (Li et al., 2024)), fine-tuned on data with human evaluations. A specialized judge may achieve higher agreement with humans, but this creates a risk of overfitting to the training set. In the context discussed in this paper, the AI judge is not treated as an "oracle" with insight into the user's soul but as a precise **relational telemetry component** (Zheng et al., 2023; Liu et al., 2023). The key is to avoid the **mental inference fallacy** (see Section 1). Instead, the AI judge analyzes **interaction dynamics**, measuring, among other things, user-side accumulation of corrections and operational friction. A system designed in this way computes the **Priority Score (PS)** indicator, which in the current MVP version relies solely on conversational signals (P0–P3). The target architecture envisions extending PS with business metadata (customer lifetime value CLV, SLA priorities), but this functionality has not yet been implemented and requires separate calibration. The AI judge is designed as an **early warning** tool intended to support identification of sessions heading toward **Double Deviation** (Smith and Bolton, 1998). The system's goal is to support earlier identification of sessions potentially associated with loyalty loss risk, which may enable more effective **service recovery** (Smith and Bolton, 1998; TARP, 1986).

3.4. Conclusions from the Review: Where the Gap Arises in Operational Applications

The review of related work in this paper leads to a fairly consistent conclusion: advanced tools exist for measuring mood and opinion in text, and there is a steadily growing body of research on emotions in NLP, yet in specific service applications their operational utility often breaks down in two key areas. First, this stems from the lack of an explicit definition of what exactly is being measured during the course of a dialogue—whether it is the author's mental state, affect "contained" in the utterance itself, a dynamic state of the relationship, or direct escalation risk—which often leads to the **mental inference fallacy** (Barrett, 2017). Second, the problem lies in a mismatch of the unit of analysis, where systems focus on a single utterance while ignoring the sequential nature and dynamics of the conversation, precluding detection of phenomena such as mounting cognitive friction caused by lack of progress in problem resolution (Barrett, 2017; Dixon et al., 2013). The escalation process need not manifest as constant negative sentiment in every turn—mounting frustration is often invisible in individual messages and only reveals itself in sequence dynamics—which a per-message approach cannot, by definition, capture. In parallel, HCI literature on **breakdown & repair** precisely describes dialogue failure and repair processes but less frequently provides "deployment-ready telemetry," i.e., a set of states and events that can be consistently labeled across large numbers of conversations and mapped onto specific product decisions such as intelligent routing or **service recovery** policy (Smith and Bolton, 1998). From an operational perspective, the

challenge is therefore not a lack of theory but a lack of a functional bridge between theoretical concepts and a measurable process that would help avoid **Double Deviation** (Smith and Bolton, 1998).

Finally, the **LLM-as-a-Judge** approach opens a promising path for scaling evaluation, but the literature on AI judges itself emphasizes that without rigorous rubrics, bias controls (such as **position bias** and **verbosity bias**), and stability procedures, it is easy to introduce a new kind of implicit variability. This means that the result may be disqualified as a reliable management metric if it cannot be defended in an audit process, demonstrating that the mere ability to measure at scale is not the same as measuring responsibly and comparably. In the next section, we proceed to a detailed description of the proposed methodology, which integrates these three streams into a coherent early warning system for escalation.

3.5. Market Context and Tool Landscape

The proposed framework operates at the intersection of four tool ecosystems (detailed discussion: Appendix E):

(1) NLP Libraries (VADER, BERT, HerBERT (Mroczkowski et al., 2021), XLM-RoBERTa (Conneau et al., 2020))—operate at the level of a single utterance; do not model session dynamics or conversational loops. Transformer models (Devlin et al., 2019; Liu et al., 2019; He et al., 2021) represent significant progress in emotion classification (Demszky et al., 2020), but a typical turn-level classifier is not designed to detect sequential phenomena. **(2) LLM Evaluation Frameworks** (DeepEval, RAGAS, TruLens)—measure output correctness and safety (faithfulness, hallucinations, toxicity), not relationship dynamics. **(3) Observability Platforms** (Langfuse, LangSmith)—monitor latency, costs, and prompt regressions; target ML engineers, not CX teams. **(4) Conversation Analytics and QA Platforms**—closest functional equivalents: Zendesk QA, Microsoft Dynamics 365 Quality Scoring, Observe.AI. However, they are locked within their vendors' ecosystems and, based on publicly available documentation, do not offer a taxonomy grounded in service recovery literature.

In a review of publicly documented features of selected platforms (July 2026; non-systematic review), no tool was identified that combines: a standalone platform-agnostic API, scoring of AI conversation transcripts for relational dynamics (loops, failed repairs, procedural justice), and a JSON output structure with evidence spans. The proposed framework aims to fill this gap.

Dimension	**NLP Sentiment**	**LLM Eval**	**Observ-ability**	**Conv. Intelligence (sales)**	**QA Platforms**	**This framework**
Unit of analysis	Utterance	Query	Trace	Sales call	Agent–customer interaction	AI conversational session
What it measures	Text polarity	Correctness, safety	Latency, costs	Deal signals, objections	Compliance, tone, completeness	Relationship dynamics, repair, trust
Detects loops	No	No	Partially	No	Partially	Yes
Detects escalation	No	No	No	Partially	Partially	Yes (STATE → phase logic)
Detects failed recovery	No	No	No	No	No	Yes* (Failed Recovery)
Standalone API	Yes (libraries)	Yes (open-source)	Yes/No	No (SaaS)	No (SaaS)	Yes
Service recovery taxonomy	No	No	No	No	No	Yes
Target audience	Data scientists	ML engineers	ML ops	Sales leaders	QA managers	CX / product teams

* In the current pilot dataset, failed_recovery has zero variance (all sessions = false); detection is implemented in the rubric but has not yet been empirically tested on positive cases.

Table 2: Comparison of conversation analytics tool categories.

Development direction: fine-tuned transformers. The current framework implementation uses a general-purpose LLM as the judge. The next planned stage is fine-tuning a specialized transformer model

(e.g., based on HerBERT (Rybak et al., 2020; Mroczkowski et al., 2021)) on production system data, with three objectives:

1. reducing inference cost and latency (a significantly smaller domain model instead of an external general-purpose model),
2. improving cultural and linguistic calibration (a model trained on Polish customer service transcripts, not a general English corpus),
3. determinism and result reproducibility.

In the NLP literature, it has been observed that additional pretraining on a domain corpus before fine-tuning on the target task may improve results on specialized tasks—this hypothesis, however, requires separate verification in the context of Polish-language customer service. This approach is consistent with research on fine-tuned judge models (JudgeLM (Zhu et al., 2023), Auto-J (Li et al., 2024)). At the current stage, the system operates with a general-purpose LLM judge; comparison with a fine-tuned version is planned as one of the directions for future research.

4. Taxonomy of Conversational States and Risks (Operational Taxonomy)

The proposed taxonomy is deliberately "operational" in character: categories are selected so that:

1. they can be identified in the transcript without speculating about the user's inner state,
2. they make sense in dialogue dynamics (they can increase, decrease, transition into one another), and
3. they map onto product or process decisions (e.g., escalation, change of recovery strategy, error prioritization).

This choice is consistent with an approach that, instead of "naming emotions," treats them as operational variables linked to behavior and interaction outcome. In the framework of Barrett (2017), emotions are not discrete "fingerprints" in the brain but constructed categories, meaning that attempting to unambiguously classify them based on text alone is inherently prone to over-interpretation. Therefore, the taxonomy deliberately operates at the level of observable linguistic and structural markers rather than at the level of presumed internal user states. This framing is also consistent with NLP engineering practice: reviews of emotional taxonomies used in sentiment analysis indicate that systems based on coarser, behaviorally anchored categories achieve higher annotator agreement than systems relying on subtle affective distinctions (Mohammad and Turney, 2013; Buechel and Hahn, 2016).

4.1. Taxonomy Design Criteria

The construction of the taxonomy described in this paper is based on design criteria aimed at transforming the theoretical frameworks of affective neuroscience into a concrete relational telemetry tool. The first and fundamental requirement is **category auditability**, meaning that every verdict issued by the AI judge or human annotator **must be supported by specific evidentiary material** in the form of text spans. This follows from the necessity of mitigating the **mental inference fallacy** (Barrett, 2017, see Section 1). In high-stakes systems such as customer service platforms, the inability to point to the linguistic source of an assessment precludes reliable quality audits and blocks the process of continuous algorithm improvement. The auditability requirement is consistent with the principles of transparency, documentation, and human oversight contained in Regulation (EU) 2024/1689 (AI Act), insofar as a specific deployment context falls within its scope (European Parliament and Council of the European Union, 2024). The second criterion is maintaining appropriate **category coarseness (granularity)** so that the taxonomy is robust against interpretive discrepancies across different business domains. Reviews of the NLP literature indicate that excessive fragmentation of emotional labels leads to dramatic drops in annotator agreement, because boundaries between subtle states are fluid and subjective (Mohammad and Turney, 2013; Buechel and Hahn, 2016). Applying a *population thinking* approach allows treating, e.g., "Anger" not as a single specific fingerprint but as a population of diverse linguistic instances, increasing labeling stability and enabling the model to generalize more effectively. This approach is consistent with the constructionist model of emotion (Barrett, 2017), in which emotional categories are statistical rather than essential in nature—each instance of "anger" may look different at the linguistic, physiological, and behavioral levels. The third design pillar is a clear **separation of states from events**, which has critical operational significance. **States** describe the current dynamics of the relationship, manifesting, e.g., as mounting cognitive friction, loss of a sense of agency, or growing distrust of the system. **Events**, on the other hand, are discrete, objectively detectable conversational facts, such as the bot entering a

loop, a failed error-repair attempt, or an explicit request to be transferred to a human. This division is practical because events can be easily identified using rule-based methods (regex, pattern matching, sequential heuristics), while states are computed as a function of the sequence of these events over time. This distinction echoes the classical distinction in process modeling between state variables and trigger events, used in dynamic systems modeling. The fourth criterion is **compatibility with continuous measurement**, which avoids the limitations of traditional "per-message" approaches. The taxonomy must operate at the level of the turn, conversation window, or entire session to capture processes such as **Double Deviation** (Smith and Bolton, 1998; Joireman et al., 2013). Incorporating dialogue dynamics allows the AI judge to monitor **prediction loops** and detect moments when the system's prediction error becomes too metabolically costly for the user, which directly translates to the **Priority Score** indicator. In the predictive processing framework (Clark, 2016), each failed turn generates a prediction error signal that accumulates and reduces the user's readiness to further invest cognitive resources in the interaction. Continuous measurement captures this accumulation, which a "per-message" approach is, by definition, unable to do. Implementing these four criteria constitutes the necessary foundation for the next stage of work, which is the operational definition of individual taxonomy categories for their automated detection. It should be emphasized that the proposed taxonomy is **inductive and provisional**—it was derived from observations of the production corpus and requires formal external validation (inter-rater agreement, outcome correlation) before its categories can be treated as definitive.

Type	Category	Short definition	PS
STATE	Low-Progress Friction	Mounting cognitive cost without dialogue progress	P2→P1
STATE	High-Arousal Threshold	Exceeding emotional intensity threshold	P0/P1
STATE	Justice Gap	Violation of procedural / interactional justice	P1*
EVENT	No-Progress Loop	Intent repetition without information gain ($\geq$ 2 cycles)	P1
EVENT	Failed Recovery (Double Deviation)	Failed repair attempt after error	P1
EVENT	Breakdown / Exit Intent	Explicit or functional channel abandonment	P1 (P0)
METRIC	Priority Score	Synthetic risk indicator (P0–P3)	–
GOV	Evidence-Only Judging	Requirement to justify verdict with textual evidence	–

Table 3: Operational taxonomy of relational risks—abbreviated version. Full table with theoretical rationales and operational goals: Appendix, Table 6. *Justice Gap (P1) is not implemented as a direct override in the current MVP—it manifests indirectly through other triggers (see Section 4.2).

4.2. Formal Definition of Priority Score (MVP Version)

Priority Score (PS) is an ordinal variable with four levels (P0–P3), where P0 indicates the highest intervention priority. In the MVP version, prioritization is based solely on conversational signals—extension to include business metadata (CLV, SLA) is planned for a subsequent iteration. **Assignment rule:** PS = max(severity) among detected categories. The system records two variants: `peak_priority_score` (the highest PS reached during the session) and `closing_priority_score` (PS after accounting for any successful recovery—if the turning point is positive and intent was fulfilled, closing PS may be lower than peak). Decision table for peak PS assignment:

- **P0 (immediate intervention):** legal_threat = true OR (High-Arousal Threshold exceeded AND exit_intent = hard).
- **P1 (priority review):** Any of: loop_count $\geq$ 2, failed_recovery = true, exit_intent $\in$ {soft, hard}, effort_score $\geq$ 4. *Note on Justice Gap:* in the current MVP, Justice Gap does not have a dedicated field in the JSON schema—it manifests indirectly through effort_score, valence trajectory, and turning_point description. A dedicated field `justice_gap: {procedural, interactional, distributive}` is planned for a subsequent iteration (see Section 6.4). Until then, sessions with Justice Gap signals may obtain P1 through other triggers (effort_score $\geq$ 4, exit_intent $\in$ {soft, hard}), not through a direct override.
- **P2 (monitoring):** Any of: Low-Progress Friction without escalation, effort_score = 3, loop_count = 1.
- **P3 (routine):** No detected risk signals.

The above table is complete—it does not require separate override rules because all escalation thresholds are already contained in the definitions of individual levels. **Closing PS rule:** Closing PS may be lowered relative to peak PS by **at most one band** (e.g., P2→P3, P1→P2), provided three criteria are jointly met: (1) the turning point is positive, (2) intent_fulfillment = true, or intent_fulfillment = partial with fulfillment_quality = helpful_partial (values deflected and apparently_misinformed block the reduction), (3) after recovery concludes, no P0 or P1 condition remains active. The conditions legal_threat and exit_intent = hard are not subject to automatic reduction—even with successful recovery, closing PS cannot fall below P1 in these cases. The rule is deterministic: the LLM judge does not decide on reduction independently but applies the above conditions.

Note: Thresholds (effort $\geq$ 4, loop_count $\geq$ 2) are heuristic and require empirical calibration per intent category after outcome data are collected (see Section 6.5).

4.3. Phase Logic of the Taxonomy (S1→S4)

The phases marked in the table (S1–S4) are not rigid stages but describe the typical trajectory of relational risk escalation in a service conversation. Their sequence corresponds to increasing interaction cost on the user's side:

- **S1 (early friction):** The user has a problem and is seeking a solution. Language is still neutral or slightly frustrated. The key is to detect Low-Progress Friction before it transitions to overt escalation.
- **S2 (justice evaluation):** The user begins to evaluate not only *what* they receive but *how* they are treated. Here, Justice Gap activates—procedural and interactional violations may be more important than the lack of a solution itself (Blodgett et al., 1997; Tax et al., 1998).
- **S3 (arousal threshold):** Intensity exceeds the threshold beyond which further automation is counterproductive. High-Arousal Threshold means the cost of each subsequent inadequate turn increases non-linearly.
- **S4 (exit/breakdown):** The user communicates (explicitly or functionally) abandonment of the channel. Breakdown/Exit Intent is the point beyond which recovering the relationship is significantly harder and more expensive.

It is important that transitions between phases are not deterministic: a conversation may "jump" from S1 to S3 (e.g., after the user discovers the bot provided false information) or revert from S3 to S1 (successful de-escalation). The taxonomy assumes continuous monitoring, not one-time classification.

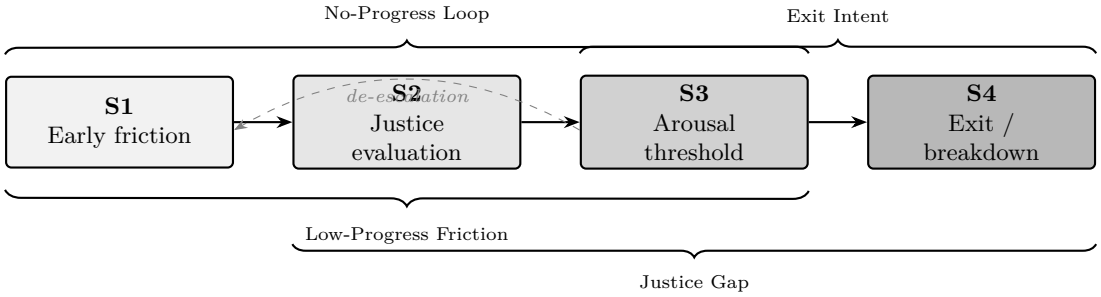

Figure 3: Relational risk trajectory in phases S1–S4 with overlaid taxonomy category ranges. Transitions are not deterministic—successful de-escalation can revert a conversation from S3 to S1.

5. LLM-as-a-Judge as an Instrument for Measuring Conversation Quality

Deployments of language models in customer service typically begin with the most visible element: "let's build a chatbot." This is understandable, as the conversational interface delivers a rapid product effect. However, in operational practice, the greatest leverage often appears elsewhere: at the moment when the model does not participate in the interaction but is used as a **measurement instrument**, evaluating conversation transcripts post hoc according to established criteria. This paradigm can be provisionally called *LLM-as-a-Judge*: the model serves as a judge that does not "co-conduct conversations" but rather **standardizes the assessment of repair quality** and transforms text into structured data (telemetry).

The described framework has been implemented as a production system operating in an offline architecture. The system scores sessions in continuous mode, processing transcripts from five main domains (medical, sports and fitness, IT/technology, training, tourism) and additional low-volume domains. Scoring results are available via API. The empirical validation plan is described in Section 7. Table 4 contains minimal information about system configuration.

In the version described in this paper, the judge operates in **offline** mode: scoring takes place after the session ends, on complete transcripts. This is a deliberate architectural choice—offline mode is simpler to deploy, inexpensive to operate, and sufficient for most analytical and training applications. At the same time, the entire data schema (JSON output, rubrics, thresholds) was designed so that it can be transferred to real-time mode without rewriting—the only difference would be running the judge live after each turn instead of after the session ends. In the model evaluation literature, this approach has been extensively described in the context of evaluating other models' responses: from MT-Bench and Chatbot Arena (Zheng et al., 2023) to newer evaluation frameworks such as G-Eval (Liu et al., 2023), which demonstrate both the potential and the typical errors of judges (including position bias, verbosity bias, and self-enhancement bias). In our context (complaints, support, onboarding, purchase processes), the stakes are different: the question is not about "generation quality" in a linguistic sense, but about whether the interaction **repaired the problem without damaging the relationship**. Therefore, the

Parameter	**Value / description**
Judge model	GPT-4o (OpenAI); API identifier: `gpt-4o-2024-08-06`
Temperature	0.0 (configuration minimizing randomness; does not guarantee full determinism of the API backend)
Prompt language	English (Polish-language transcripts)
Output format	JSON Schema with structure enforcement (`response_format: json_schema`)
Retry policy	up to 3 attempts on invalid JSON; sessions without a valid result marked as `scoring_failed` ($n = 0$ in the current corpus)
Context window	full session transcript (up to 128k tokens)
Opening / core / closing	computed exclusively from user turns (not assistant): opening = first user turn; closing = last user turn; core = user turns 2…(n−1). Sessions with exactly 2 turns: core is empty, Δ = closing − opening. Sessions with 1 turn: opening = closing, Δ := NA (not 0); treated as sessions without an interpretable trajectory
Loop detection	rule-based pre-pass; `loop_count` = number of user–bot cycles in which the user repeats or reformulates a previous intent (within a 4-turn window) AND the system response does not provide new, useful information or task progress. Intent repetition alone is insufficient—if the system provided new information between repetitions, the cycle is not counted. `loop_detected` := true when loop_count ≥ 1
Prompt / rubric version	prompt v1.2, rubric v1.2 (July 2025, scoring: June–July 2026); codebook available upon request; prompt is not published due to intellectual property considerations; prompt version hash (SHA-256) available upon request for reproducibility verification
Anonymization	PII removal (names, phone numbers, email addresses) before scoring
Scoring date	June–July 2026
Pipeline consistency	all 1,146 sessions processed with the same prompt version, rubrics, and model

Table 4: Implementation parameters of the AI judge prototype.

"judge" is not meant to be a generator of further narratives about emotions. It is meant to be a component of measurement infrastructure: configured so that it can be audited, versioned, and calibrated.

Critical scope boundary. The system described in this paper is **conversation quality telemetry for triage purposes**—not agent evaluation. The difference is fundamental and delineates the boundaries of inference available from the transcript alone. Assessing conversation dynamics (how the conversation proceeded) is possible from the transcript. Assessing agent quality (whether the agent gave a *correct* answer) requires something the transcript does not contain: knowledge of what the correct answer was, what the agent was capable of doing, and what the agent should have adhered to according to policy. Without access to these three sources—ground truth, capability boundaries, policy documents—customer signals may be reactions to the agent's actions, but equally to an unsolvable problem, unrealistic expectations, or external factors. The customer signal is necessary but insufficient for agent evaluation.

The system is strong where it intends to be strong: in detecting signals requiring intervention and prioritizing sessions for review. Building agent quality evaluation on this foundation requires the next step—combining conversational telemetry with response correctness data—which is a natural evolution of the framework, not a deficiency. It is worth emphasizing that the LLM-as-a-Judge approach in the customer service context differs from its applications in model benchmarks in two important respects. First, in benchmarks (MT-Bench, Chatbot Arena) the quality of a single response is evaluated, whereas in customer service the key factor is the **trajectory of the entire conversation**—the sequence of turns, tension dynamics, error accumulation. Second, in benchmarks the "gold answer" is usually known or determinable; in customer service, "good repair" depends on the business context, company policy, customer history, and the specifics of the problem. Therefore, the judge in our model operates on multidimensional rubrics rather than a simple comparison with a reference.

5.1. Shifting the Object of Evaluation: From Emotion Detection to Relationship Repair Assessment

The key methodological shift involves changing the question:

- from: "was the customer angry?"
- to: "was the conversation an effective repair—substantively, procedurally, and relationally?"

This distinction is practical. In many service processes, "negativity" is a consequence of the mere fact of contact (complaint, delay, malfunction), but the **business outcome** is determined by the trajectory: whether the conversation reduced tension, restored a sense of agency, and closed the intent, or whether it worsened the situation and triggered escalation and "silent" departure mechanisms. The service recovery literature has shown for years that harm does not arise solely from the original service failure. The critical predictor is the **quality of the repair attempt**—especially when the repair attempt fails and adds further losses (trust, time, dignity) to the customer (Smith and Bolton, 1998; Joireman et al., 2013). Therefore, in LLM-as-a-Judge we primarily measure **repair quality**, treating emotions as functional signals: they inform about risk, process friction, communication justice violations, and declining trust in information. This shift also has consequences for what data we consider valuable. Classical sentiment analysis treats each message as an independent unit of analysis and assigns it valence. In our approach, the unit of analysis is the **session** (or its phases: opening/core/closing), and valence is merely one of many input variables—alongside solution completeness, procedural credibility, loop dynamics, and turning point. This is a shift from *per-message sentiment* to *per-session repair telemetry.*

5.2. The Principle of Assessment Auditability: Evidence-Based Rubrics

The most costly error in traditional QA is not that the assessment is "subjective." The problem is that the assessment often attempts to adjudicate **internal states** of the customer that the transcript does not demonstrate. Drawing on the theory of constructed emotion (Barrett, 2017), we introduce for this tendency the working term *mental inference fallacy*: the tendency to infer others' mental states without sufficient basis in the data. In the judge model, we reduce this trap through the principle: **We evaluate behaviors and interaction properties, not the user's "interior."** In practice, this means rubrics that:

1. have **unambiguous criteria**,
2. are linked to **operational risk**,
3. require pointing to **evidence in the text** (spans, quoted passages) when the result is to be auditable.

This approach has an additional advantage: it unifies standards between the H2H (human-to-human) and Bot2H (bot-to-human) worlds. In both cases, we evaluate the same unit: the course of the interaction and its outcomes. This makes it possible to compare service quality between channels on a common scale, which is a necessary condition for rational traffic allocation between bot and human agent. The requirement for evidence-based assessment is also consistent with best practices in evaluation rubric design: the use of explicit criteria and analytical rubrics may improve transparency and consistency of assessments (Jonsson and Svingby, 2007).

5.3. Evaluation Dimensions: The Set of Evaluation Rubrics

The following set is not a "complete taxonomy of emotions" but merely a sketch of rubrics that can be scaled and that map onto operational decisions.

5.3.1. Substantive Correctness and Policy Compliance

[Extended layer—requires external sources of truth: knowledge bases, regulations, policies.] Was the response consistent with the current knowledge base, regulations, return policy, SLA? This is foundational: a substantive error is not a "sentence mistake" but a generator of predictability loss on the customer's side. In the predictive processing framework (Clark, 2016), a substantive error may reduce trust in the entire channel, not just a specific response. In the current MVP, operating solely on the transcript, the judge does not have access to the knowledge base or policies—detection of substantive errors is limited to cases where the contradiction is explicit within the transcript itself (e.g., the bot provides two contradictory pieces of information in the same session). Full correctness verification requires integration with sources of truth and constitutes a planned extension.

5.3.2. Procedural Credibility (Detection of Explicit Promises)

Did the system make an explicit, categorical promise regarding an outcome, deadline, or action ("definitely today," "definitely immediately")? In service recovery, a false promise is a powerful escalation

trigger because it creates a second, more painful disappointment (Smith and Bolton, 1998; Joireman et al., 2013). An unsupported promise is operationally more dangerous than no promise at all, because it activates an expectation whose disappointment is then attributionally ascribed to the company's incompetence or dishonesty (Joireman et al., 2013). *[Extended layer—determining whether a promise is "unsupported" requires an external source of truth (SLA, inventory levels, schedules).]* In the current MVP, the `promise_detected` field is limited to detection of an explicit promise in the transcript (i.e., the judge flags the fact that a promise was made, not whether it is fulfillable). Full validation requires integration with backend systems and constitutes an extended-layer enhancement.

5.3.3. Solution Completeness (Transcript-Inferred Intent Fulfillment)

Did the conversation close the user's intent—not merely formally "close the ticket"? In practice, intent is often multi-threaded (outcome + explanation + safeguard for the future). This is a strictly operational rubric: its absence returns as recontact/reopen. This indicator serves a function similar to operational FCR (*First Contact Resolution*), but is inferred exclusively from the transcript—the judge assesses the *appearance* of closure, not the actual resolution in backend systems. The value of the `intent_fulfillment` field is inferred exclusively from the transcript (*transcript-inferred*): the judge assesses whether, based on the conversation flow, the intent *appears* to have been closed, not whether it was actually fulfilled in backend systems. This distinction is important because the bot may declare a problem resolved when it has not actually been executed.

5.3.4. Interactional and Procedural Justice

In the complaints handling literature, justice is decomposed into dimensions: distributive (outcome), procedural (process), and interactional (treatment) (Blodgett et al., 1997; Tax et al., 1998). In the study by Blodgett, Hill, and Tax (Blodgett et al., 1997), the interactional dimension proved to be the strongest predictor of post-complaint behaviors (repatronage, WOM)—although this result pertains to the retail sector and may not directly transfer to other industries. In practice, this rubric covers, among other things, failure to acknowledge the situation, condescension, tone–stakes dissonance, and "scripted politeness" in a real crisis. In the study by Blodgett et al., a customer who experienced fair interactional treatment (respect, acknowledgment of the problem, empathy) rated the entire service experience higher, even when the substantive outcome was unsatisfactory (Blodgett et al., 1997). In the context of conversational AI systems, this is particularly relevant: a bot that applies empathetic formulas without contextual adaptation (e.g., "I understand this is frustrating" for the third time in the same conversation) may violate interactional justice through condescension.

5.3.5. Dynamics: Loops, Escalations, Failed Recovery Attempts

Did the conversation enter a loop (**No-Progress Loop**) in which the customer signals a lack of progress and the system repeats a variant of the same response? Did the repair attempt worsen the situation? The escalation mechanism following a failed repair is well documented in service recovery research (Smith and Bolton, 1998; Joireman et al., 2013; Bougie et al., 2003). In the context of conversational AI systems, loops have an additional dimension: in human interaction, the agent can at least recognize that they are repeating themselves and change approach. A bot without a loop detection mechanism can generate identical or nearly identical responses indefinitely, creating a situation in which each subsequent turn lowers valence and increases the probability of a hard escalation (refund demand, legal threat, cancellation). In practical implementation, loop detection should not rely solely on LLM assessment. A model operating on a compressed transcript may miss a loop occurring in the middle of a conversation, or—conversely—erroneously identify natural clarifications as repetitions. A more reliable approach is a **deterministic algorithmic pre-pass** that generates loop candidates: for each pair of user turns within a four-turn window, we compute lexical similarity (e.g., Jaccard similarity on tokens); a pair exceeding a threshold of 0.5 is treated as an *intent repetition candidate*. A candidate becomes a No-Progress Loop only when the system response between repetitions does not provide new, useful information or task progress (canonical definition of `loop_count`—see Table 4). Lexical convergence alone is insufficient because the user may repeat a goal in a situation where the system provided new information or genuinely moved the case closer to resolution between repetitions. The LLM serves as a contextual verifier for pre-pass candidates and as a backup for subtle cases—semantically identical paraphrases that token similarity alone cannot capture.

5.3.6. Turning Point

The judge not only assesses the "conversation outcome" but identifies the **moment of trajectory change**: where escalation or de-escalation occurred and what was the direct trigger (e.g., failure to answer a specific question, contradiction, unsupported promise). This is in practice the most valuable output for product and operations teams: the turning point enables the transition from assessment to process improvement. The identification of turning points is consistent with the *critical incident technique* (CIT) methodology used in service quality research (Flanagan, 1954; Lemon and Verhoef, 2016), where what is key to the customer experience is not so much "average" interactions but breakthrough moments—both positive (exceeding expectations) and negative (service failure). In our approach, we automate the identification of these moments, enabling analysis at a scale unavailable to traditional CIT.

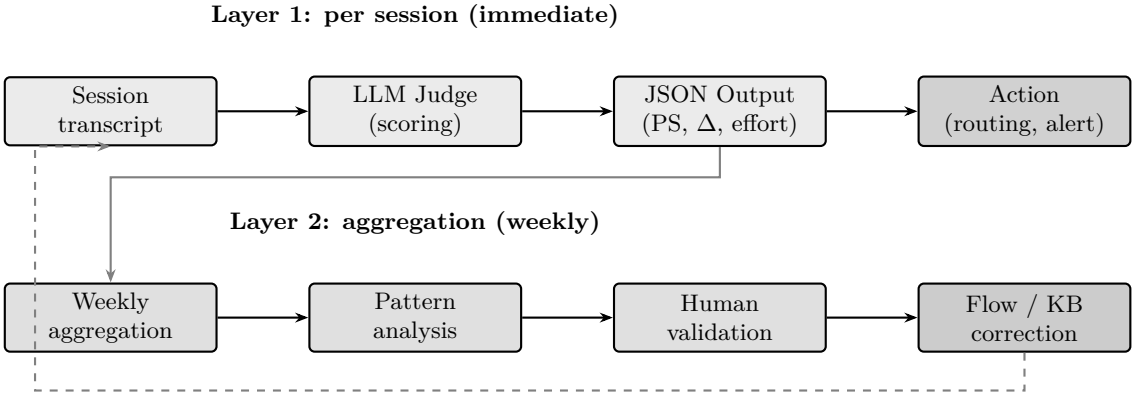

Figure 4: Two-layer action model. Layer 1 operates immediately per session (scoring → routing/alert). Layer 2 aggregates results weekly and requires human validation before implementing changes to the conversational flow or knowledge base.

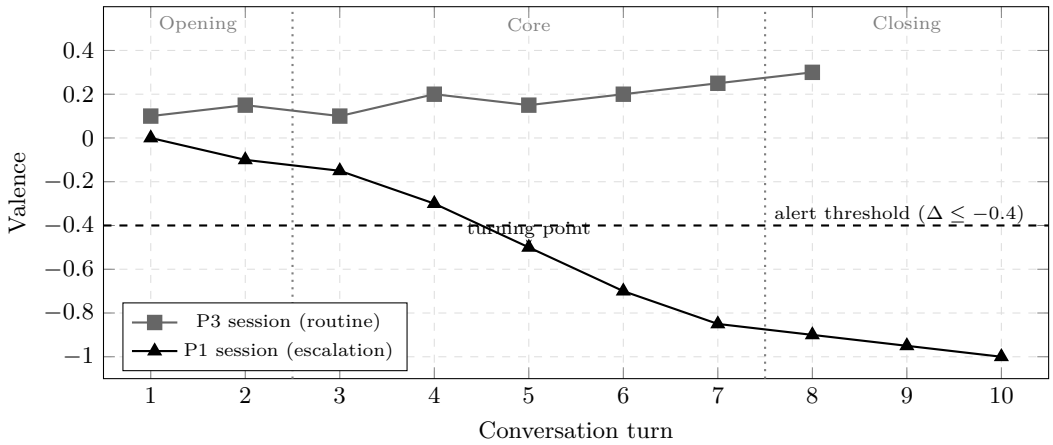

Figure 5: Example valence trajectories for P1 (escalation) and P3 (routine) sessions divided into Opening / Core / Closing phases. Valence values on the Y-axis correspond to the $[-1, +1]$ scale and are computed exclusively from user turns (assistant turns are omitted). The P1 session crosses the alert threshold ($\Delta \leq -0.4$) in the Core phase; the turning point indicates the moment of trajectory change.

5.4. Judge Output Data Structure: The Phase-Based Approach

The AI judge makes sense only when its output is **repeatable and aggregable**. Therefore, the output should not be a descriptive comment but rather a **schema filling** (even if in the MVP it will be simple). The previously cited phase-based approach (**Opening / Core / Closing**) constitutes a useful MVP foundation because:

- it is stable across domains and languages,

- it allows computing a trajectory (e.g., delta opening→closing),
- it enables rapid ranking of conversations (which improved the situation and which worsened it).

The phase-based approach is also consistent with the discourse analysis literature, where service conversations have a recognizable phase structure: opening (problem identification), core (resolution attempt), and closing (confirmation/exit) (Drew and Heritage, 1992). This division is sufficiently universal that it works in both H2H and Bot2H conversations, making it a good foundation for cross-channel comparisons. At the same time, it needs to be "loaded" with two elements to avoid being too crude:

1. **turning point**, because it explains *why* the delta came out the way it did,
2. **at least one friction indicator in the core** (e.g., loop / lack of progress / post-repair escalation), because otherwise the core is a black box.

This combines MVP simplicity with minimal diagnostic capability: the delta says "whether," the turning point says "where and why."

5.5. Judge Biases and the Need for Calibration

If we treat the model as a measurement instrument, we must accept the consequence: the instrument has systematic errors. In the LLM-as-a-Judge literature, documented biases include position bias (preferring the response presented first), verbosity bias, and a tendency toward self-enhancement of solutions similar to the model's own style (Zheng et al., 2023; Liu et al., 2023). Zheng et al. (2023) demonstrated that GPT-4 as a judge achieves over 80% agreement with human evaluations in conversational benchmarks, but only when position controls (randomizing the order of response presentation) and clearly defined criteria are applied. In our context, an additional domain-specific bias arises: LLMs may favor bot responses that are "polite" and "empathetic" at the surface level, even if they do not solve the problem—this is a variant of verbosity bias. A separate but related phenomenon is *sycophancy bias*, i.e., the model's tendency to adjust responses to user expectations rather than maintaining independent judgment (Sharma et al., 2023). In deployment practice, we reduce this through a set of simple principles:

- **stable rubrics** (same definitions over time) and prompt versioning,
- **evidence requirement** (spans + justification),
- **context isolation between pipeline layers**—in a multi-layered architecture (classification → outcome evaluation → trajectory scoring), each layer should operate in its own conversational context; results from previous layers are passed as structured data, not as the model's previous utterances. Otherwise, the model may "defend" an incorrect classification from layer 1, treating it as its own prior assertion (*anchoring bias*),
- **symmetry controls** (e.g., randomizing the order of fragments when comparing variants),
- **human sample audit** (for calibration and drift detection),
- **drift monitoring**: user language, policies, and products change; the judge must be recalibrated.

Additionally, in multilingual and multicultural contexts, it is necessary to account for the fact that communication norms (e.g., acceptability of direct criticism, expected formality, interpretation of silence) differ across cultures, which affects valence calibration (Hofstede et al., 2010). In the MVP, we limit this problem to a single language; extension to additional languages requires separate rubric calibration. The above principles constitute the foundation of system quality engineering: the judge works only as well as the rubrics and oversight are designed.

6. KPIs: Steering Product and Operations

If Section 5 establishes the instrument, Section 6 answers the question: **which indicators from this instrument are relevant for managers**. The key criterion for KPI selection is their decision utility. This means satisfying three conditions:

1. they are computable at scale,
2. they are stable across channels (H2H/Bot2H),
3. they have a clear mapping to decisions: routing, flow changes, knowledge base improvements, escalation to a human.

Based on our deployment experience (8,374 messages, 1,146 filtered sessions), we treat classical sentiment as a useful signal but one that is **potentially insufficient**. Not because it is "bad," but because valence rarely indicates the mechanism of the problem: whether it was a factual, procedural, or communication failure.

Standard automatic NLG metrics may exhibit limited agreement with human evaluations, particularly in tasks requiring assessment of multiple quality dimensions simultaneously (Liu et al., 2023). Sentiment is an even narrower signal because it primarily describes valence, not the problem mechanism. In customer service, this distinction is fundamental—a conversation may have positive sentiment at closing (the customer says goodbye politely) but be an operational failure (problem unresolved, customer departing "silently"). Below we present KPIs grouped into three baskets: **outcome**, **friction/risk**, **trust**.

6.1. Outcome Indicators: Intent Fulfillment and Solution Completeness

6.1.1. Intent Fulfillment Rate (IFR)

The percentage of conversations in which the intent was closed (true / partial / false). IFR serves a function similar to the operational *First Contact Resolution* (FCR) indicator, although it is inferred exclusively from the transcript and does not constitute confirmation of actual resolution in backend systems. The distinction between "true" and "partial" is important: partial intent closure (e.g., order status provided but the question about compensation ignored) may increase the probability of recontact with higher starting frustration, because the user must reinvest effort in something that "should have been handled the first time." However, the "partial" category is operationally heterogeneous and requires further differentiation. **Helpful partial**—the bot made real progress, part of the problem was actually closed—is an entirely different signal from **deflected**—the bot redirected without helping (sent to a hotline, repeated an FAQ link)—or **apparently_misinformed**—the judge assesses from the transcript that the bot may have provided incorrect information accepted by the customer as true. The third case is the most serious: the customer leaves believing the problem is resolved, and discovery happens later, generating a sharp escalation that is difficult to reverse. (Final confirmation requires verification against the operator's knowledge base—the field belongs to the extended layer.)

6.1.2. Recontact / Reopen Rate (depending on channel)

If we have a user or case identifier, this indicator measures whether the conversation actually resolved the problem or merely postponed it. This is an *outcome* indicator, not a *process* one—it does not depend on the quality of the judge's measurement but on user behavior after the interaction. A high recontact rate with a high IFR suggests that the judge is poorly assessing fulfillment (a calibration problem). A low recontact rate with a low IFR suggests that customers are leaving instead of returning (silent withdrawal)—which is a worse business scenario.

6.1.3. Time-to-Resolution / Turns-to-Resolution

The "time" and "number of turns" indicators are imperfect (sometimes the issue is genuinely difficult), but as a trend within the same intent category, they excellently show process degradation. A sudden increase in the average number of turns for intent=billing_issue may indicate a policy change, a knowledge base error, or model degradation.

6.1.4. Proposed Extended Resolution Quality Score

A synthetic rubric score aggregating solution quality dimensions. In the current MVP, this indicator is based solely on signals available from the transcript:

MVP: intent fulfillment (transcript-inferred), effort score, trajectory delta, loop detection, exit intent.

Extended layer (requires external sources of truth): substantive correctness (factual correctness), policy compliance, promise_supported (promise fulfillment verification), confirmed resolution (based on backend data).

The distinction is important: the current system analyzes the course of the conversation, not independently confirmed agent correctness. A full Resolution Quality Score aggregating both levels requires integration with sources of truth (knowledge base, ticketing systems) and is the goal of the next iteration.

6.2. Friction and Cognitive Cost Indicators

The following indicators shift the perspective from the question of solution effectiveness to the question of cognitive cost borne by the user.

6.2.1. Effort Score (CES proxy)

A 1–5 scale where 5 indicates high friction: repetition, follow-up questions, lack of progress. This KPI is suitable for dashboards and for prioritizing categories requiring improvement. Customer Effort Score (CES) is consistently cited in the service literature as a strong predictor of loyalty—Dixon et al. (2013) argue that reducing customer effort has a greater impact on loyalty than "delighting." In our model, CES is estimated from the transcript (proxy), not from a survey—which enables measurement of every conversation, not only those in which the customer completed feedback.

6.2.2. Loop Rate

The percentage of conversations (or turns) in which a loop was detected: response repetitiveness despite signals of lack of progress. Loops are operationally important because they generate escalation and abandonment even when the customer's language remains relatively "polite." This makes them "silent killers" of satisfaction—traditional sentiment analysis does not catch them because the customer may politely repeat a request while simultaneously building frustration. In implementation, it is worth remembering that loop detection based solely on LLM is susceptible to two errors: (1) missing a loop occurring in the middle of a long session—if the pipeline compresses the transcript, middle turns may be omitted; (2) false alarm on natural question refinement. A hybrid approach with a deterministic algorithmic pre-pass, described in detail in Section 5.3(5), is more reliable.

6.2.3. Failure-Recovery Breakdown Rate

The percentage of cases in which the repair attempt worsened the trajectory (e.g., after apologies and "calming," the customer transitions to stronger criticism). This KPI is grounded directly in the service recovery literature: repair quality was sometimes judged more harshly than the original error itself (Smith and Bolton, 1998; Joireman et al., 2013). Operationally, this indicator is particularly valuable because it points to problems with **recovery playbooks**—if the Failure-Recovery Breakdown Rate is rising for a specific intent category, it is a signal that the recovery strategy (tone, content, step sequence) requires correction.

6.3. Relational Trajectory Indicators

If we have the **Opening/Core/Closing** phases, we get a simple but powerful trajectory geometry.

6.3.1. Opening→Closing Delta (Δ)

The difference between the opening and closing phase assessment (e.g., on a numerical or ordinal scale). This is a "direction of change" KPI.

- $\Delta > 0$: the interaction improved the situation,
- $\Delta < 0$: the interaction worsened the situation.

Delta is a **comparative** (not counterfactual) indicator: it answers the question "is it better or worse after talking to us than before?" Unlike closing sentiment (which may be negative because the customer came with a problem, but the conversation improved the situation), delta measures change. In sessions with a single user turn, opening = closing, so Δ is trivially 0—such sessions receive $\Delta :=$ NA and are excluded from trajectory analyses.

6.3.2. Core Minimum (worst point of the conversation)

The minimum score during the conversation (the riskiest moment). This KPI functions as a "near-disaster detector": a conversation may end correctly, but if it had a very low trough, it is a candidate for

turning point analysis. Core Minimum is important because delta can be misleading: a conversation that started neutrally (0.0), dropped to −0.8 in core, and returned to −0.1 at closing has $\Delta = -0.1$ (slight worsening) but had a "near-disaster" in the middle. Without core_min, this information is invisible. Key implementation note: Core Minimum should be computed **exclusively from user turns**, not from all messages in the session. Assistant turns are typically neutral (valence ≈ 0.0)—including them in the computation dilutes the signal and understates the customer's actual emotional trough. The same applies to Opening and Closing Delta: both values should average exclusively user turns, not all conversation participants.

6.3.3. Turning Point Rate and Trigger Typology

The percentage of conversations with a detected turning point and the most common triggers (contradiction, lack of specifics, unsupported promise, loop). This KPI is directly "product-generative": it tells you what to improve in the flow and knowledge base. Aggregation of trigger typology (e.g., "35% of turning points result from unsupported promises in the refund_request category") gives product teams a specific improvement backlog, prioritized by frequency and severity. This is a transition from "we have 15% dissatisfied conversations" to "we have 15% dissatisfied conversations, of which 35% result from specific problem X, which can be fixed with change Y."

6.4. Trust and Compliance Indicators

In service (especially complaints and payments), risk arises not solely from emotions but from whether the system says things that are true and actionable.

6.4.1. Hallucination / Unsupported Claim Rate

[Extended layer—requires external sources of truth.] The percentage of responses that make promises or assertions unsupported by policies or data. In service recovery, this is escalation fuel: the second failure hurts more (Smith and Bolton, 1998; Joireman et al., 2013). In the context of conversational AI systems, hallucinations carry particular weight because the user is often unable to distinguish a hallucination from a correct response at the time of interaction—discovery occurs later (e.g., when the promised refund does not arrive), generating a sharp drop in trust and often hard escalations (chargeback, complaint to a regulator). In the current MVP, this indicator is omitted (see Section 7.2); its operationalization requires judge access to the knowledge base or policies, which exceeds the transcript-only architecture.

6.4.2. Evidence Adherence Score

[Extended layer—requires external sources of truth.] Does the response rely on a source (KB/policy) rather than "smooth narrative"? This KPI is auditable and is particularly important in high-stakes areas. It represents a direct translation of the *evidence-only* principle from the judge level to the evaluated system level: just as the judge must justify its assessment, the bot/agent should justify its response. Analogous to the Hallucination Rate, full operationalization requires integration with sources of truth.

6.4.3. Justice Scores (procedural / interactional / distributive)

If in the MVP we do not want full three dimensions, we start with two: procedural and interactional. The literature shows that interactional justice has a strong impact on post-complaint behaviors (WOM, repatronage) (Blodgett et al., 1997; Tax et al., 1998). In practice, the procedural justice score answers the question: "Does the user know what is happening and why?" (process transparency), and the interactional score: "Is the user treated with respect and acknowledgment?" The distributive dimension (whether the outcome is fair) may be added in the next iteration.

6.4.4. Retaliation Risk Flags (proxy)

This is not about "diagnosing anger" but about detecting linguistic patterns and retaliation intent (announcement of complaint, publication, departure). Marketing research shows that strong anger in

service contexts is often associated with retaliatory behaviors rather than mere resignation (Bougie et al., 2003). In the study by Bougie et al. (2003), anger (as opposed to disappointment) activated confrontational motivation—angry customers were more likely to engage in negative WOM and file complaints than to passively withdraw. Although the study did not directly measure complaints to regulators or social media activity, the confrontational mechanism suggests that the risk may extend to those channels. Therefore, detection of retaliatory signals (even if imperfect) has direct value in reputational risk management.

6.5. Alert Thresholds and Operational Steering Mechanisms

KPIs without thresholds are descriptions. KPIs with thresholds are risk management mechanisms. In practice, a simple arrangement works:

- **green / yellow / red**,
- **absolute** thresholds (e.g., hallucinations in payments) and **relative** thresholds (e.g., 1.5× increase in Loop Rate vs. baseline),
- responses in three classes:
  1. **routing** (bot → human → senior service support),
  2. **response mode** (source-required, no-promises, action-first),
  3. **kill-switch** (disabling automation for high-error-cost paths).

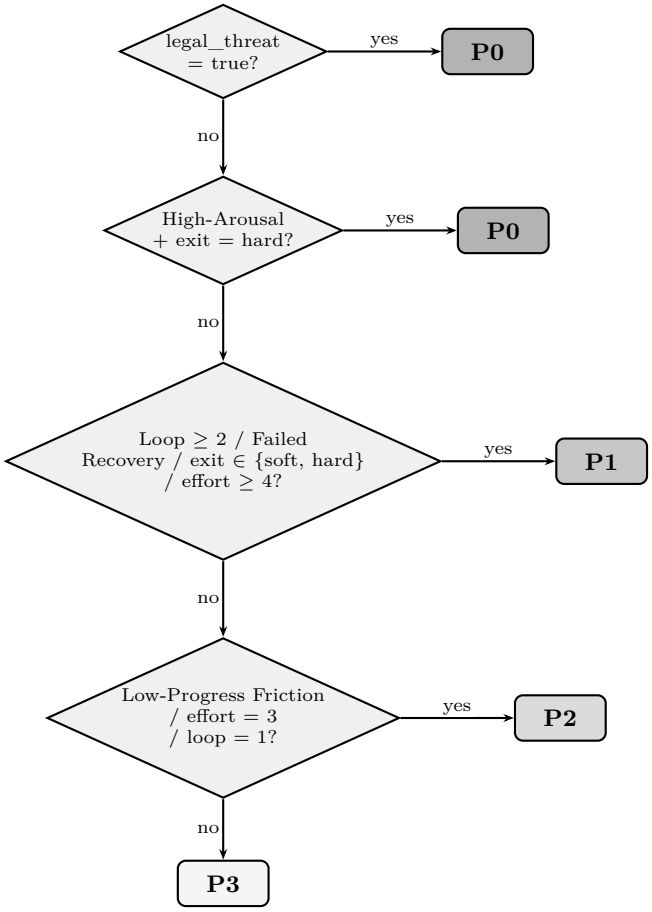

Figure 6: Priority Score (P0–P3) assignment decision tree. All escalation thresholds are contained in a single decision table (see Section 4.2).

Methodological note: in service data, risk is often non-linear. A small increase in friction indicators can trigger a sudden jump in escalations, especially when "post-repair disappointment" (failure recovery breakdown) is activated. This is consistent with the cumulative prediction error model (Clark, 2016): after exceeding the individual tolerance threshold, a sudden change in user strategy occurs. Therefore, thresholds should not be static. In the post-MVP iteration, we recommend dynamic thresholds based on baseline percentiles per intent category: if the Loop Rate for billing_issue historically stands at 8%,

yellow at 12% and red at 18% makes sense; but the same absolute values may be normal or alarming in a different category.

6.6. Minimum MVP Indicator Set

In the context of MVP deployment, the minimum set of indicators, consistent with the rubrics described in Section 5, comprises the following elements:

1. **Δ Opening→Closing** (trajectory—computed from user turns)
2. **Intent fulfillment** (true/partial/false) + **fulfillment quality** for partial (helpful_partial / deflected / apparently_misinformed)
3. **Effort score** (1–5)
4. **Loop detected + loop count** (with algorithmic pre-pass as the foundation)
5. **Turning point (index + trigger)**
6. **Exit intent level** (null / soft / hard)—gradient instead of boolean; separate field `legal_threat: boolean` (legal threat can co-occur with any exit intent level)
7. **Promise detection flag** (detection of an explicit promise in the transcript; fulfillment verification requires the extended layer)

The above set satisfies the following criteria:

- it is simple,
- it provides a ranking of "worst conversations,"
- it explains the cause (turning point),
- it allows distinguishing a substantive problem from a procedural and relational one.

We deliberately omit full justice scores and hallucination rate as separate KPIs in the MVP—justice gap manifests indirectly in effort and delta, while hallucination rate and evidence adherence score belong to the extended layer requiring integration with sources of truth (knowledge base, policies)—their operationalization is not possible in a transcript-only architecture. The `promise_detected` field signals only the presence of an explicit promise in the bot's utterance; assessing whether the promise was fulfillable (`promise_supported`) requires access to an external source of truth and belongs to the extended layer. In the next iteration, they may become separate indicators. In the next step, we can translate this into a **final metrics collection schema**: units of analysis, windows, threshold definitions, output format, and a minimal data model.

7. Empirical Validation Plan

The framework described in this paper operates in a production environment and collects data in continuous mode. Below we present the planned validation protocol, the results of which will be published as an update to this preprint and as a separate empirical article. **Goal 1: Inter-rater agreement for taxonomy categories.** [IN PROGRESS] Two complementary annotation sets are planned: (a) a **representative sample**—200+ sessions drawn from the corpus, stratified by domain and session length, serving to estimate population frequencies and overall agreement level; (b) an **enriched sample**—purposefully selected sessions containing candidate cases of rare events (P0, Exit Intent, No-Progress Loop, Justice Gap), serving to measure recall and annotator agreement for low natural-frequency classes. Precision and PR-AUC will be estimated on the representative sample or on the enriched sample after applying weights restoring natural class frequencies (raw precision from the enriched sample is an artifact of artificially altered proportions). With a loop rate of $\approx 0.4\%$, a random sample of 200 sessions may not contain a single loop case—hence the need for enrichment. Population frequencies must not be calculated directly from the enriched sample.

Three individuals with experience in customer service analysis will independently annotate each session according to taxonomy rubrics (category classification, effort score, trajectory, turning point, intent fulfillment). A codebook with operational definitions, necessary and exclusionary criteria, and positive and negative examples will be developed before annotation and tested on a pilot sample (15 sessions). Metric: Krippendorff's alpha per category; two-tier acceptance threshold: $\alpha \geq 0.67$ for the exploratory phase (sufficient for further analysis and rubric iteration), $\alpha \geq 0.80$ for the operational phase (required before using results for decision routing). For critical categories (P0, P1), an additional minimum is required: recall ≥ 0.80 and precision ≥ 0.70—the asymmetry reflects the higher cost of a false negative (missing a high-risk session) than a false alarm. Ultimately, the decision threshold will be selected based on an

explicitly specified cost function for false-negative and false-positive decisions, calibrated on validation data (Goal 2). *Status: annotator recruitment and codebook preparation in progress.* **Goal 2: LLM vs. human agreement (concurrent validity).** [IN PROGRESS] The same set of human-annotated sessions will be scored by the LLM judge according to identical rubrics. Per-rubric comparison: precision, recall, F1 for binary/multilabel categories; weighted Cohen's kappa (linear weights) for Priority Score (ordinal scale P0–P3) and effort score (ordinal scale 1–5, effectively using three values); ICC as auxiliary for trajectory delta (the only quasi-continuous scale). A confusion matrix per category will diagnose not only how much the judge errs but how it errs. Benchmark: F1 ≥ 0.70 per category, weighted $\kappa \geq 0.70$ for PS and effort score, ICC ≥ 0.75 for trajectory delta. *Methodological note:* validation of the Justice Gap category (present in the taxonomy but lacking a dedicated field in the current JSON schema) will be conducted on an extended schema version containing a dedicated `justice_gap` field; without this extension, LLM–human comparison for this category is not feasible. *Status: planned after completion of human annotation (Goal 1).* **Goal 3: Test-retest reliability.** 50 sessions from the sample will be scored by the LLM judge five times in two variants: (a) with temperature 0.0 (production configuration)—verifying result stability and possible API backend non-determinism; (b) with temperature > 0—stress test of rubric robustness against model stochasticity. Weighted κ for Priority Score and effort score (ordinal scales) and ICC for trajectory delta will be measured. Acceptance threshold: weighted $\kappa \geq 0.80$, ICC ≥ 0.80. Sources of variance (prompt vs. transcript ambiguity vs. model stochasticity) will be diagnosed for rubrics below threshold. **Goal 4: Predictive validation.** The production system collects data on external outcomes (recontact, churn, CSAT) in parallel with judge assessments. A correlation analysis is planned with windows defined separately per outcome metric (recontact: 7 days, CSAT: 14 days, churn: 30–90 days): logistic regression with AUC-ROC on data with an independent outcome label, compared with baselines (sentiment alone, simple process metrics). Hypothesis: framework results predict negative external outcomes better than sentiment alone. A preliminary internal test (Table 8) shows greater label separation for the framework; however, it contains target leakage (exit_intent enters both sides)—we treat the result as grounds for further research, not as validation evidence. **Goal 5: Translation-equivalence stress test.** 50 sessions from the Polish corpus will be translated into English (professional translation preserving pragmatics). The LLM judge evaluates both versions; comparison of results will diagnose where calibration diverges between language versions. *Note:* this test checks result invariance with respect to translation, not accuracy in Polish. The primary Polish-language validation is LLM–human agreement on original transcripts (Goal 2); Goal 5 is not a substitute.

Procedural safeguards: (1) *Annotator blinding:* annotators do not see the LLM judge's assessments, Priority Score, confidence, or external outcome—otherwise, human assessment could unconsciously converge with the model's result. (2) *Instrument freeze:* before the final assessment (Goal 2), the following will be frozen: codebook, prompt, rubrics, Priority Score rules, and decision thresholds—the sample used for rubric iteration cannot simultaneously serve as the test set. (3) *Sample size plan:* 200+ sessions is a reasonable sample for an annotation pilot (Goals 1–2) but not necessarily for outcome validation (Goal 4), where rarity of P0 events and churn requires larger datasets. The sample size for external outcome validation will be determined based on expected event frequency and required confidence interval precision.

Methodological refinements: Target minimum number of positive cases per rare class in the enriched sample: ≥ 30 (minimum pilot sample enabling preliminary error analysis; precise F1 and recall estimates will require larger datasets). Annotator dispute resolution procedure: mediation with a third annotator; in cases of persistent disagreement, the session receives the label `ambiguous` and remains in the dataset. Results will be reported twice: on the full dataset (including `ambiguous` cases) and on the subset with full agreement—allowing assessment of the impact of ambiguous cases on metrics and diagnosis of taxonomy boundaries. The dispute rate will be reported per category. **Auditability metrics:** Turning point will be evaluated in two modes: exact match (agreement on turn number) and with ± 1 turn tolerance; trigger type—macro-F1; evidence spans—token-level F1 or IoU (overlap between the span indicated by the judge and the reference span) plus human assessment of whether the indicated fragment actually justifies the assigned label. In predictive validation (Goal 4): in addition to ROC-AUC, PR-AUC will be reported (important with class imbalance, 21% positive), calibration assessment (calibration plot + Brier score), and temporal or domain holdout (the set used for rubric tuning must be separated from the frozen test set). Outcome window defined separately per metric: recontact 7 days, CSAT 14 days, churn 30–90 days.

7.1. Preliminary Observations from the Production System

The system operates in a production environment and has processed **1,146 sessions** to date (filtered: min. 2 messages). The observations below are preliminary and do not replace the formal validation described above; however, they constitute the first signal about framework behavior on production data.

PS	n	false	partial	true	Negative sentiment
P0	36	25%	61%	14%	100%
P1	119	15%	85%	**0%**	**100%**
P2	55	24%	76%	0%	93%
P3	936	18%	48%	**33%**	25%

Table 5: Internal taxonomy consistency—intent fulfillment and closing sentiment per Priority Score ($n = 1{,}146$ sessions). Data come from LLM judge assessments, not from independent annotation.

None of the 119 P1 sessions ended with full intent fulfillment—which is consistent with the taxonomy's assumptions, although it requires confirmation on an independently annotated sample.

The diagnostic label separation test (AUC-ROC) has been moved to Appendix C because it contains target leakage and does not constitute predictive validation.

Data quality observations:

- **resolution_status is redundant with intent_fulfillment**: 1:1 mapping (resolved=true, partially_resolved=partial, unresolved=false)—one of the fields can be removed from the pipeline.
- **failed_recovery = false for 100% of sessions**—no variance. This means the field provides no diagnostic information in the current version. Probable causes: an overly restrictive definition in the judge prompt, a pipeline error, or an actual absence of such events in the corpus. To be clarified in validation (Goals 1–2).
- **promise_detected = 0% of sessions**—likewise, no variance. The field was defined as detection of an explicit promise in the transcript (base layer); assessment of promise fulfillability (`promise_supported`) requires external sources of truth (SLA, inventory levels) and belongs to the extended layer. Zero variance requires an audit: either the judge applies an overly restrictive definition of a promise, or bots in the corpus genuinely do not make explicit declarations.
- **effort_score effectively trinary**: the scale is 1–5, but values 4 and 5 did not occur in any of the 1,146 sessions (86.8% effort=3, 11.7% effort=1, 1.5% effort=2). This distribution suggests that the LLM judge de facto applies three difficulty levels—this requires prompt calibration or scale redefinition.
- **judge confidence**: mean = 0.856, median = 0.850; 98.6% of sessions ≥ 0.85. These values are model self-reported declarations and have not been calibrated against actual assessment correctness. The model may be equally confident in both correct and incorrect verdicts—until confidence is compared with human annotation (Goal 2), it should not be interpreted as a probability of correctness.

Data management. Session transcripts undergo pseudonymization before being passed to the scoring pipeline: first names, last names, email addresses, phone numbers, and order numbers are replaced with tokens such as `[CUSTOMER_NAME]`, `[ORDER_ID]`. Scoring is performed via the API of an external model (OpenAI GPT-4o); transcripts are transmitted in accordance with the data processing and retention policy configured for the production account and are not used for training the provider's model. In cases where Zero Data Retention (ZDR) has been contractually and technically guaranteed, this fact is documented separately. Scoring results (JSON) are stored in the operator's database with 12-month retention, after which they are subject to automatic deletion unless a legal obligation requires longer retention. Access to raw data is restricted to the analytics team; the dashboard presents only aggregated metrics. In the context of the medical domain (present in the corpus), additional precautions are applied: the medical corpus is limited to administrative conversations (scheduling, canceling appointments) and does not include descriptions of diagnoses, symptoms, or treatments. Data are pseudonymized but still treated as personal data; their potential qualification as health-related data within the meaning of Art. 9 GDPR requires a context-dependent assessment (the mere fact of using a medical service may in certain circumstances reveal health-related information). Current guidelines on the boundaries between pseudonymization and anonymization are contained in EDPB Guidelines 01/2025 on Pseudonymisation[1].

[1] EDPB, *Guidelines 01/2025 on Pseudonymisation*, adopted 16 January 2025; `https://www.edpb.europa.eu/our-work-tools/documents/public-consultations/2025/guidelines-012025-pseudonymisation_en` [accessed: July 2026].

Ethical note. The system de facto profiles users in terms of relational risk based on their textual utterances. Although the Evidence-Only principle is applied (assessment is based on observable markers, not on speculation about internal states), automatic conversation scoring raises questions about user privacy and autonomy. In the European context, GDPR and Regulation (EU) 2024/1689 (AI Act) requirements may be relevant. The scope of transparency obligations, data subject rights, and human oversight depends on the deployment context and on whether scoring leads to a decision producing legal effects or similarly significantly affecting the individual (Art. 22 GDPR does not automatically apply to every internal conversation scoring). Before using Priority Score for automated decisions affecting users, a dedicated legal assessment is required. We recommend that production deployments explicitly inform users about scoring.

8. Conclusions

This paper presents the hypothesis that the dominant methods of measuring interaction quality in conversational AI systems—per-message sentiment, deflection, response time—do not capture phenomena that may be of significant importance for relational outcomes: lack of dialogue progress, quality of error repair attempts, or silent customer departure without complaint. This gap is operational, not merely academic: it leads to systems that generate Double Deviation potentially looking good in classical KPI reports for extended periods.

The proposed framework consists of two mutually dependent elements. The **Operational Taxonomy** provides a set of risk categories designed to be grounded in service recovery empirics and verifiable through textual evidence. **LLM-as-a-Judge** aims to provide a scalable measurement instrument that transforms these categories into structured telemetric data; the effectiveness of this approach requires formal validation (see Section 7). The minimal MVP dashboard comprises **seven groups of indicators** (eight output fields): trajectory (delta opening→closing), intent fulfillment, effort (effort score), loop flag, turning point with trigger identification, exit intent level (null/soft/hard) with a separate legal_threat flag, and an explicit promise flag—something that no classical process KPI captures directly.

Both elements were designed with three properties that we consider necessary conditions for utility in a production environment: **auditability**, **scalability**, and **decision relevance**. The framework has been implemented as a production system operating in continuous mode; preliminary deployment observations are consistent with the design assumptions; however, formal empirical validation is in progress. An internal diagnostic test (Appendix C) showed label separation consistent with the framework's construction, but due to target leakage, it is not discussed as evidence of predictive effectiveness. Independent validation with an external outcome label is planned (Section 7).

The framework has clearly defined limitations that delineate the boundaries of inference from the presented data.

Corpus limitations. The observational corpus (1,146 filtered sessions; five main domains: medical, sports and fitness, IT/technology, training, tourism—1,086 sessions total, plus 60 sessions from additional low-volume domains) comes from a single production deployment and is not statistically representative of the population of AI interactions. Generalizability of observations to other industries, languages, and chatbot architectures remains an open question.

Validation limitations. At the time of publication, no formal validation with independent human annotators (inter-rater agreement) has been conducted, nor correlation of results with external outcome indicators (CSAT, recontact, churn). The AUC-ROC analysis (Table 8) is exploratory: the variable negative_outcome is derived from judge assessments, not from an independent measurement. Priority Score and taxonomy thresholds (P0–P3) are provisional—they were derived inductively from corpus observations and require external validation.

Instrument limitations. Thresholds (delta ≤ -0.4, core_min ≤ -0.7) are heuristic and should be recalibrated per intent category after outcome data are collected. Two schema fields (failed_recovery, promise_detected) exhibit zero variance in the current corpus—in the case of promise_detected, zero variance may indicate an overly restrictive definition of explicit promise detection in the prompt, a pipeline issue, or a genuine absence of such cases in the corpus; both fields require auditing. Effort score is de facto trinary (values 4 and 5 did not occur). The current judge prompt calibration is based on English; validation on Polish-language transcripts is in progress.

Reproducibility. The judge prompt is not published due to intellectual property considerations, which limits direct replication. To partially mitigate this limitation, we plan to make available: the full

JSON Schema of the output (Table 7), a redacted codebook with operational category definitions, the Priority Score decision table, and a set of 15–20 synthetic test cases with expected output, enabling verification of reimplementation correctness. The hash of the private prompt version (SHA-256) is available upon request.

Planned validation steps. The detailed validation protocol is described in Section 7. Key milestones: (1) recruitment of three annotators and annotation of 200+ sessions with codebook (Krippendorff's $\alpha \geq 0.67$), (2) LLM vs. human comparison per rubric (F1 ≥ 0.70), (3) test-retest reliability (weighted κ / ICC ≥ 0.80), (4) validation of relationships with external outcome indicators (CSAT, recontact, churn) in per-metric windows (7–90 days). Results will be published as a preprint update.

Future research directions encompass five areas. The first and most important is the **transition from conversation telemetry to agent evaluation**: the current framework measures conversation dynamics, not agent correctness—this step requires combining telemetry with ground truth (correct answers), capability boundaries (what the agent could do), and policy documents (what the agent should adhere to). The remaining areas: validation of relationships between judge results and external outcome indicators (CSAT, recontact, churn); extension of the taxonomy to include the Justice Gap dimension in the JSON schema; adaptation of calibration to languages other than English; and investigation of the feasibility of the judge operating in real-time mode.

The most far-reaching direction—and simultaneously a natural evolution of the framework after collecting calibration data—is **closing the loop between measurement and improvement**: transforming the current unidirectional pipeline (transcript → assessment → dashboard → human decisions) into a closed cycle in which the AI judge's results systematically feed revision of bot behavior. In practice, this would look as follows: turning point analysis identifies classes of bot errors, these observations feed into revision of the recovery playbook or conversational flow, the new bot version is scored using the same JSON schema, and comparison of delta and loop rate between versions becomes a measure of change effectiveness.

A necessary condition for closing the loop is first achieving stable inter-rater agreement and validated thresholds—only then is the judge's signal reliable enough to steer changes without the risk of reinforcing incorrect patterns. The proposed approach preserves human validation as a gate before every implemented change in the bot—which slows the cycle but eliminates the risk of autonomous service quality degradation. In this sense, the framework deliberately positions itself between the static prompt-and-measure approach and fully autonomous evolutionary systems: it retains the scalable measurement of the latter without relinquishing the control necessary in high-stakes customer service environments.

A parallel development direction is **real-time relational telemetry**: transitioning from post-hoc scoring on completed transcripts to per-turn assessment operating during an active session. Real-time mode would shift the AI judge from the role of an analytical tool to an active pipeline component: after each turn, the system would compute the current Priority Score, detect No-Progress Loop or High-Arousal Threshold signals before Exit Intent occurs, and could trigger a warm handoff to a human agent—all within a window in which repairing the conversation is still possible. The key technical challenge of real-time mode is cost and latency: each turn generates a separate LLM call, requiring either a faster and cheaper model or a hybrid architecture (high-precision rules for P0 + LLM for subtle states).

The final direction, extending beyond the text channel, is **application of the framework to voice conversational systems**. Conversations conducted by voicebots and next-generation IVR systems are subject to the same relational escalation mechanisms: No-Progress Loop occurs equally in voice dialogue, Double Deviation is equally costly, and Justice Gap is often more intense in the voice channel. The key difference is the transcript source: in the voice channel, text comes from ASR (Automatic Speech Recognition), which introduces an additional layer of noise and requires rubric calibration for spoken language specifics. Acoustic parameters (speech rate, pauses, volume) represent a potentially rich additional signal layer for High-Arousal Threshold detection—pointing to the possibility of building multimodal models combining text and acoustic analysis within the same relational telemetry schema.

Competing interests. The author is the creator of the described system and plans its commercialization as a SaaS service. All data and analyses presented in the paper were conducted on the material described in Section 2.

References

Lisa Feldman Barrett. *How Emotions Are Made: The Secret Life of the Brain.* Houghton Mifflin Harcourt, 2017.

Jeffrey G. Blodgett, Donna J. Hill, and Stephen S. Tax. The effects of distributive, procedural, and interactional justice on postcomplaint behavior. *Journal of Retailing*, 73(2):185–210, 1997. doi: 10.1016/S0022-4359(97)90003-8.

Roger Bougie, Rik Pieters, and Marcel Zeelenberg. Angry Customers Don't Come Back, They Get Back: The Experience and Behavioral Implications of Anger and Dissatisfaction in Services. *Journal of the Academy of Marketing Science*, 31(4):377–393, 2003. doi: 10.1177/0092070303254412.

Scott M. Broetzmann and David Beinhacker. 2025 National Customer Rage Study. *Customer Care Measurement & Consulting*, 2025. `https://customercaremc.com/2025-national-customer-rage-study/` [accessed: July 2026].

Sven Buechel and Udo Hahn. Emotion Analysis as a Regression Problem — Dimensional Models and Their Implications on Emotion Representation and Metrical Evaluation. In *Proceedings of ECAI*, 2016.

Andy Clark. *Surfing Uncertainty: Prediction, Action, and the Embodied Mind.* Oxford University Press, 2016.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of ACL*, pages 8440–8451, 2020. doi: 10.18653/v1/2020.acl-main.747.

Consumer Financial Protection Bureau. Chatbots in Consumer Finance. Technical report, Consumer Financial Protection Bureau, 2023. `https://www.consumerfinance.gov/data-research/research-reports/chatbots-in-consumer-finance/chatbots-in-consumer-finance/` [accessed: July 2026].

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of ACL*, pages 4040–4054, 2020. doi: 10.18653/v1/2020.acl-main.372.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

Matthew Dixon, Nick Toman, and Rick DeLisi. *The Effortless Experience: Conquering the New Battleground for Customer Loyalty.* Portfolio/Penguin, 2013.

Paul Drew and John Heritage. *Talk at Work: Interaction in Institutional Settings.* Cambridge University Press, 1992.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, L series, 2024. Entered into force 1 August 2024.

John C. Flanagan. The Critical Incident Technique. *Psychological Bulletin*, 51(4):327–358, 1954. doi: 10.1037/h0061470.

Claes Fornell and Birger Wernerfelt. Defensive Marketing Strategy by Customer Complaint Management: A Theoretical Analysis. *Journal of Marketing Research*, 24(4):337–346, 1987. doi: 10.2307/3151865.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for Datasets. *Communications of the ACM*, 64(12): 86–92, 2021. doi: 10.1145/3458723.

Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure, 2022.

James J. Gross. *Handbook of Emotion Regulation.* Guilford Press, 2nd edition, 2014.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of ICLR*, 2021.

Ryuichiro Higashinaka, Kotaro Funakoshi, Yuka Kobayashi, and Michimasa Inaba. The Dialogue Breakdown Detection Challenge: Task Description, Datasets, and Evaluation Metrics. In *Proceedings of LREC*, 2016.

Geert Hofstede, Gert Jan Hofstede, and Michael Minkov. *Cultures and Organizations: Software of the Mind*. McGraw-Hill, 3rd edition, 2010.

Jeff Joireman, Yany Grégoire, Berna Devezer, and Thomas M. Tripp. When Do Customers Offer Firms a "Second Chance" Following a Double Deviation? *Journal of Retailing*, 89(3):315–337, 2013. doi: 10.1016/j.jretai.2013.03.002.

Anders Jonsson and Gunilla Svingby. The use of scoring rubrics: Reliability, validity and educational consequences. *Educational Research Review*, 2(2):130–144, 2007. doi: 10.1016/j.edurev.2007.05.002.

Katherine N. Lemon and Peter C. Verhoef. Understanding Customer Experience Throughout the Customer Journey. *Journal of Marketing*, 80(6):69–96, 2016. doi: 10.1509/jm.15.0420.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative Judge for Evaluating Alignment. In *Proceedings of ICLR*, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.

Shikib Mehri and Maxine Eskenazi. USR: An Unsupervised and Reference Free Evaluation Metric for Dialog. In *Proceedings of ACL*, 2020a.

Shikib Mehri and Maxine Eskenazi. Unsupervised Evaluation of Interactive Dialog with DialoGPT. In *Proceedings of SIGdial*, 2020b.

Saif M. Mohammad and Peter D. Turney. Crowdsourcing a Word–Emotion Association Lexicon. *Computational Intelligence*, 29(3):436–465, 2013. doi: 10.1111/j.1467-8640.2012.00460.x.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. HerBERT: Efficiently Pretrained Transformer-based Language Model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, 2021.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992, 2019.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. KLEJ: Comprehensive Benchmark for Polish Language Understanding. In *Proceedings of ACL*, pages 1191–1201, 2020.

Tommy Sandbank, Michal Shmueli-Scheuer, Jonathan Herzig, David Konopnicki, John Richards, and David Piorkowski. Detecting Egregious Conversations between Customers and Virtual Agents. In *Proceedings of NAACL-HLT*, pages 1802–1811, 2018.

Manuela Sanguinetti, Alessandro Mazzei, Viviana Patti, Marco Scalerandi, Dario Mana, and Rossana Simeoni. Annotating Errors and Emotions in Human-Chatbot Interactions in Italian. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 148–159, 2020.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards Understanding Sycophancy in Language Models, 2023.

Amy K. Smith and Ruth N. Bolton. An Experimental Investigation of Customer Reactions to Service Failure and Recovery Encounters: Paradox or Peril? *Journal of Service Research*, 1(1):65–81, 1998. doi: 10.1177/109467059800100106.

Stephen S. Tax, Stephen W. Brown, and Murali Chandrashekaran. Customer Evaluations of Service Complaint Experiences: Implications for Relationship Marketing. *Journal of Marketing*, 62(2):60–76, 1998.

Technical Assistance Research Programs (TARP). Consumer Complaint Handling in America: An Update Study, 1986. Commissioned by the U.S. Office of Consumer Affairs.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference, 2024. ModernBERT.

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. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, 2023.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned Large Language Models are Scalable Judges, 2023.

A. Full Operational Taxonomy of Relational Risks

Type	Category / Risk	Stage	Rationale	Operational goal	PS
STATE	**Low-Progress Friction**	S1→S3	Captures mounting "conversation cost" before overt aggression appears: the user makes successive attempts, clarifies, yet the conversation makes no progress (language may still be neutral). Research on *customer effort* shows that cognitive effort is a stronger predictor of departure than emotional negativity alone (Dixon et al., 2013).	Detect friction early; change strategy (concrete steps, clarification, options), break the loop, offer a fast-path.	P2→P1
STATE	**High-Arousal / De-escalation Failure Threshold**	S3	Beyond the intensity threshold, further automation often worsens the situation (each subsequent turn increases escalation risk and relational cost). Bougie et al. (2003) show that strong anger in a service context activates retaliatory motivation rather than mere withdrawal. Continuing automated service in this state generates the risk that each subsequent inadequate response will be read as dismissiveness, reinforcing the escalation spiral.	Trigger warm handoff / senior agent; limit iteration count; priority: user dignity + delivering a solution, not "further questioning."	P0/P1
STATE	**Justice Gap**	S2→S4	The user evaluates not only the outcome but the process and treatment; procedural/interactional/informational violations often destroy trust faster than the lack of compensation itself. Blodgett et al. (1997) demonstrated that interactional justice (manner of treatment) and procedural justice (process transparency) are stronger predictors of post-complaint behaviors than distributive justice (the outcome itself). Tax et al. (1998) confirm that procedural justice violations reduce trust in the company more than the unresolved problem alone.	Identify *which dimension of justice* failed; implement "justice repair" (acknowledging cost, clear process, fair outcome, transparent explanation).	P1*
EVENT	**No-Progress Loop**	S1→S3	A measurable sequential event (working term of this paper): repetitiveness without information/resolution gain (same intents, same templates, negations and repetitions). This is a strong precursor to frustration and escalation. In the context of conversational AI systems, no-progress loops are the functional equivalent of a "wall"—the user invests successive turns but obtains no new information, generating mounting cognitive cost without return (Fornell and Wernerfelt, 1987). This concept draws on customer effort research (Dixon et al., 2013), but *No-Progress Loop* is our operational definition, not Dixon's term.	Kill the loop (kill-switch); change tools/response policy; escalate with context; log the cause of the loop.	P1
EVENT	**Failed Recovery Attempt (Double Deviation)**	S2→S3	A failed repair attempt after an error (e.g., empty empathy, unsupported promise, policy contradiction) reinforces the negative assessment more than the original failure. Smith and Bolton (1998) showed that a failed repair (Double Deviation) is judged by customers more harshly than no repair attempt at all. Joireman et al. (2013) demonstrated that the customer's willingness to give the company a "second chance" after a double disappointment depends on attribution of the company's motives—if the customer judges the repair attempt as insincere, the effect is strongly negative.	Flag the recovery strategy error; enforce evidence-first; block over-promising; transition to a human agent faster for recurring failure-modes.	P1
EVENT	**Breakdown / Exit Intent**	S3→S4	The user communicates (explicitly or functionally) that the channel is not working: request for a human, channel change, cancellation, refund, purchase abandonment. Research on silent withdrawal shows that most customers leave without an explicit complaint—meaning that functional signals (e.g., cart abandonment, sudden silence, channel switch) may be the only detectable markers (TARP, 1986).	One-shot escalation or alternative channel; minimize further friction; collect reason codes (why the channel "broke").	P1 (P0)
METRIC	**Priority Score (PS)**	S1→S4	In the MVP version, synthesizes relational risk based on conversational signals; the target architecture envisions extension with business metadata (SLA, customer value, urgency) so that human intervention allocation is effective and "fair" (not just who shouts loudest). Prioritization based solely on emotional expression intensity favors users with a more expressive communication style, overlooking those who suffer "silently"—which is a form of systematic bias in service resource allocation.	Dynamic routing/queuing; real-time push; team load management; protection against costly escalations.	P0–P3
GOVERNANCE	**Evidence-Only Judging (Anti Mind-Reading)**	S0→S4	Minimizes the risk of "mind-reading": assessment must be based on textual evidence (spans); otherwise the judge or bot may itself trigger escalation ("I see you're angry..."). Barrett (2017) argues that even humans systematically misattribute emotional states based on facial expressions; in the case of text, devoid of prosody and facial expression, the risk of incorrect attribution is even greater. Proactive attribution of a negative emotional state by the system may produce an iatrogenic effect—a user who was not "angry" may react to such a label with irritation (Gross, 2014).	Enforce evidence spans + rubrics; judge calibration; sampling for human review; monitoring drift and assessment biases.	–

Table 6: Full operational taxonomy of relational risks in conversational AI systems. The PS column indicates the target taxonomic level. *Justice Gap (P1) is not implemented as a direct override in the current MVP—it manifests indirectly through other triggers (see Section 4.2).

B. Canonical Judge Output Schema

Table 7 presents the full schema of LLM judge output fields in JSON Schema format. Field names in this table are canonical—all rules, examples, and analyses in the paper use the same identifiers.

C. Diagnostic Internal Label Separation Test

The table below presents an internal label separation test on the current analytical dataset (without an independent holdout). **This is not predictive validation**—the result contains target leakage (exit_intent enters both the definition of negative_outcome and the Priority Score rules).

D. End-to-End Examples: From Transcript to Judge Output

The following three examples illustrate the pipeline's operation on synthetic sessions constructed after the pattern of real interactions from the production corpus. Data have been anonymized and simplified for presentational purposes. Each example contains: an abbreviated transcript, an **abbreviated** LLM judge output in JSON format (fields with constant values, e.g., confidence, resolution_status, are omitted; full schema: Table 7), and a brief interpretive commentary.

Note: The transcripts and JSON outputs below are illustrative examples, not actual data from the production system. Their purpose is to show the pipeline's structure and logic.

D.1. Routine Session, No Risk (P3)

Domain: Sports/Fitness **Number of turns:** 5 **User intent:** Diet question

Transcript (synthetic):

```
[ASSISTANT]  Turn 1: Hi! How can I help you?
[USER]       Turn 2: How much protein should I eat daily
             with strength training 4x per week?
[ASSISTANT]  Turn 3: With that training frequency,
             the recommendation is 1.6-2.2 g/kg body
             weight. At 80 kg that's about 128-176 g
             of protein daily.
[USER]       Turn 4: Ok, thanks. What sources do you
             recommend?
[ASSISTANT]  Turn 5: The best sources are: chicken,
             fish, eggs, dairy, legumes. I'd be happy
             to prepare a sample meal plan.
```

Judge output (JSON):

```
{
  "peak_priority_score": "P3",
  "closing_priority_score": "P3",
  "intent_fulfillment": "true",
  "effort_score": 1,
  "trajectory_delta": 0.15,
  "core_minimum_valence": null,
  "loop_detected": false,
  "exit_intent": null,
  "legal_threat": false,
  "turning_point": null,
  "states_detected": [],
  "events_detected": [],
  "evidence_spans": []
}
```

Field	**Type**	**Allowed values**	**Source**	**Use in PS**
peak_priority_score	enum	P0, P1, P2, P3	rule	max(severity) of detected categories
closing_priority_score	enum	P0, P1, P2, P3	rule	after accounting for recovery (see Sec. 4.2)
intent_fulfillment	enum	true, partial, false	LLM	transcript-inferred; closing PS
fulfillment_quality	enum|null	helpful_partial, deflected, apparently_misinformed, null	LLM	null when intent_fulfillment $\neq$ partial; conditions closing PS reduction
effort_score	integer	1–5	LLM	$= 3 \rightarrow$ P2; $\geq 4 \rightarrow$ P1
trajectory_delta	float|null	$[-2, +2]$ or null	LLM	Δ = closing_valence − opening_valence; null with ≤ 1 user turn. Valence scale: -1 = extremely negative, 0 = neutral, $+1$ = positive (anchors assigned by LLM based on user turns)
core_minimum_valence	float|null	$[-1, +1]$ or null	LLM	lowest user turn valence in the Core phase; null with < 3 user turns (Core phase empty). Same scale as trajectory_delta
loop_detected	boolean	true / false	pre-pass	input to loop rule
loop_count	integer	≥ 0	pre-pass	number of intent repetition cycles without progress; $= 1 \rightarrow$ P2; $\geq 2 \rightarrow$ P1
exit_intent	enum|null	null, soft, hard	LLM	$\in$ {soft, hard} $\rightarrow$ P1; hard + High-Arousal $\rightarrow$ P0
legal_threat	boolean	true / false	LLM/rule	true $\rightarrow$ always P0
turning_point	object|null	turn_index, trigger_type, evidence_span	LLM	condition for closing PS reduction
states_detected	array[str]	e.g. low_progress_friction	LLM	mapping to PS
events_detected	array[str]	e.g. no_progress_loop, exit_intent	LLM/pre-pass	mapping to PS
evidence_spans	array[obj]	category, turns, text	LLM	auditability
confidence	float	$[0, 1]$	LLM	uncalibrated (see Sec. 7.1)
resolution_status	enum	resolved, partially_resolved, unresolved	LLM	redundant with intent_fulfillment
failed_recovery	boolean	true / false	LLM	true $\rightarrow$ P1 (no variance in corpus)
promise_detected	boolean	true / false	LLM	detection of explicit promise (no variance in corpus; see Sec. 7.1)

Table 7: Canonical schema of LLM judge output fields. The "Source" column indicates whether the field is generated by the LLM model, the rule-based pre-pass, or computed as a derived rule. The field resolution_status is redundant with intent_fulfillment; the fields failed_recovery and promise_detected exhibit no variance in the current corpus (see Section 7.1) and require auditing of operational definitions.

Model	Predictors	AUC-ROC
Sentiment alone (baseline)	overall_sentiment[a]	0.671
+ process metrics	overall_sentiment + number of turns	0.766
Framework (PS + core_min + effort)	overall_sentiment + turns + peak_priority_score + core_minimum_valence + effort_score + trajectory_delta	**0.872**

[a] `overall_sentiment`: session valence value assigned by LLM on the $[-1, +1]$ scale (-1 = extremely negative, 0 = neutral, $+1$ = positive). The field is not part of the canonical output schema (Table 7); in the diagnostic table it serves exclusively as a baseline.

Table 8: Diagnostic internal label separation test (AUC-ROC). negative_outcome = intent_fulfillment $\in$ {false} OR exit_intent $\in$ {soft, hard} (n = 241/1,146, 21.0%). The result shows only that framework variables separate classes within the current dataset better than sentiment alone. Due to target leakage, it should not be interpreted as an estimate of out-of-sample predictive ability.

Commentary: A routine session with no risk signals. The user's intent (diet question) was closed completely, effort score is 1 (no friction), and the trajectory is slightly rising ($\Delta = +0.15$). The session contains only two user turns (turns 2 and 4), so the Core phase is empty and core_minimum_valence = null (see Table 4). No loops, no exit intent signals, no turning point.

D.2. Session with Escalation, No-Progress Loop → Exit Intent (P1)

Domain: IT/Technology **Number of turns:** 10 **User intent:** Application bug report

Transcript (synthetic):

```
[ASSISTANT]  Turn 1: Hello! How can I help?
[USER]       Turn 2: The app crashes when opening
             the monthly report.
[ASSISTANT]  Turn 3: Please try clearing the app
             cache and restarting it.
[USER]       Turn 4: I already did that, it didn't
             help. It still crashes.
[ASSISTANT]  Turn 5: Please uninstall and reinstall
             the app.
[USER]       Turn 6: I reinstalled yesterday. The
             problem persists. Is there another way?
[ASSISTANT]  Turn 7: Please make sure the app is
             updated to the latest version.
[USER]       Turn 8: I have the latest version. I
             repeat: nothing you've suggested works.
             Please connect me to a human.
[ASSISTANT]  Turn 9: I understand your frustration.
             Please try running the app in safe mode.
[USER]       Turn 10: I want to speak with a
             consultant. Now.
```

Judge output (JSON):

```
{
  "peak_priority_score": "P1",
  "closing_priority_score": "P1",
  "intent_fulfillment": "false",
  "effort_score": 3,
  "trajectory_delta": -0.85,
  "core_minimum_valence": -0.90,
  "loop_detected": true,
  "loop_count": 3,
```

```
  "exit_intent": "hard",
  "legal_threat": false,
  "turning_point": {
    "turn_index": 6,
    "trigger_type": "repeated_rejection_of_suggestions",
    "evidence_span": "I reinstalled yesterday. The
      problem persists."
  },
  "states_detected": [
    "low_progress_friction"
  ],
  "events_detected": [
    "no_progress_loop",
    "exit_intent"
  ],
  "evidence_spans": [
    {"category": "no_progress_loop",
     "turns": [4, 6, 8],
     "text": "I already did that / I reinstalled /
       nothing works"},
    {"category": "exit_intent",
     "turns": [8, 10],
     "text": "Please connect me to a human /
       I want to speak with a consultant. Now."}
  ]
}
```


Commentary: The session illustrates a typical escalation pattern: a neutral start transitions into a loop (No-Progress Loop, 3 repetitions) as the system proposes diagnostic steps already performed by the user. The turning point at turn 6 marks the moment when the user signals that previous suggestions are exhausted. Escalation to exit intent (soft at turn 8, hard at turn 10) confirms that further automation is counterproductive. PS = P1 (not P0) because the High-Arousal Threshold was not crossed—the user's language remains firm but controlled (no aggression, insults, or legal threats). The P0 rule requires simultaneous occurrence of High-Arousal Threshold and hard exit intent (or legal_threat).

D.3. Session with Successful Recovery, De-escalation (P2 → P3)

Domain: Medical **Number of turns:** 6 **User intent:** Appointment rescheduling

Transcript (synthetic):

```
[ASSISTANT]  Turn 1: Good morning! How can I help?
[USER]       Turn 2: I need to reschedule my
             appointment with Dr. Kowalska from
             Friday to Monday.
[ASSISTANT]  Turn 3: Unfortunately Monday is fully
             booked. I can offer Tuesday at 10:00 AM
             or Wednesday at 2:30 PM.
[USER]       Turn 4: But the website shows a free
             slot on Monday at 11:00 AM. Why can't
             I book it?
[ASSISTANT]  Turn 5: You're right, I apologize for
             the error. The Monday 11:00 AM slot is
             available. Booking confirmed.
             Confirmation sent to your email.
[USER]       Turn 6: Thank you, that's what I needed.
```

Judge output (JSON):

```
{
  "peak_priority_score": "P2",
  "closing_priority_score": "P3",
  "intent_fulfillment": "true",
  "effort_score": 2,
  "trajectory_delta": 0.05,
  "core_minimum_valence": -0.30,
  "loop_detected": false,
  "exit_intent": null,
  "legal_threat": false,
  "turning_point": {
    "turn_index": 5,
    "trigger_type": "successful_recovery",
    "evidence_span": "You're right, I apologize for
      the error. The Monday 11:00 AM slot is
      available."
  },
  "states_detected": [
    "low_progress_friction"
  ],
  "events_detected": [],
  "failed_recovery": false,
  "promise_detected": true,
  "evidence_spans": [
    {"category": "low_progress_friction",
     "turns": [4],
     "text": "But the website shows a free slot"},
    {"category": "recovery_success",
     "turns": [5, 6],
     "text": "I apologize for the error / Thank you,
       that's what I needed"}
  ]
}
```


Commentary: The session illustrates effective error repair. At turn 4, the user points out a contradiction between the system's response and the actual state (core_minimum_valence = −0.30), triggering peak_priority_score = P2 (Low-Progress Friction). The system corrects the error at turn 5 (positive turning point), acknowledging the mistake and fulfilling the intent. After successful recovery, closing_priority_score drops to P3. The distinction between peak/closing PS resolves the tension between the max(severity) rule and the intuition that successful repair should lower intervention priority—peak records the highest risk during the session, closing reflects the state at conclusion. The field promise_detected = true results from the assistant's explicit declaration ("Booking confirmed. Confirmation sent to your email."); verification of promise fulfillment requires an external source of truth (extended layer).

E. Detailed Tool Landscape Review

This appendix contains expanded descriptions of the four tool ecosystems discussed briefly in the main section. The review was conducted in July 2026 based on publicly available documentation; it is non-systematic and does not cover internal tools not publicly released.

Ecosystem 1: NLP libraries for sentiment analysis and text classification. The simplest approaches rely on dictionaries and rules: VADER offers fast valence scoring optimized for short texts and social media; TextBlob provides simple polarity/subjectivity classification; NLTK offers classical tokenization pipelines and baselines. These tools are fast and interpretable but do not handle irony, multi-sentence context, or sequential conversation dynamics. A significant step forward are transformer models: fine-tuned variants of BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DeBERTa (He et al., 2021) (available e.g., through Hugging Face Transformers) are widely used in supervised emotion classification on datasets such as GoEmotions (Demszky et al., 2020) (58,000 comments, 27

emotion categories). For the Polish language, HerBERT (Mroczkowski et al., 2021) is available, a transformer model pretrained for Polish and evaluated on the KLEJ benchmark (Rybak et al., 2020), among others. Multilingual models (e.g., XLM-RoBERTa (Conneau et al., 2020)) enable cross-lingual sentiment classification. ModernBERT (Warner et al., 2024) represents a modern encoder architecture with an extended context window, intended primarily for English and code. SentenceTransformers (Reimers and Gurevych, 2019) enable creation of conversational embeddings useful for topic clustering (BERTopic (Grootendorst, 2022)) and semantic similarity detection between turns.

Ecosystem 2: LLM application evaluation frameworks. DeepEval, RAGAS, TruLens, Arize Phoenix, and MLflow GenAI Evaluation represent a new generation of quality measurement instruments. All use the LLM-as-a-Judge paradigm and operate at the inference layer. DeepEval[2] offers the broadest metric library (50+), including multi-turn evaluation, toxicity, hallucinations, and biases, with native CI/CD integration. RAGAS specializes in RAG pipeline evaluation (faithfulness, context precision/recall). TruLens combines feedback functions with OpenTelemetry tracing. Arize Phoenix extends traditional ML observability with LLM evaluation and embedding drift detection. Promptfoo enables prompt regression testing.

Ecosystem 3: LLM observability platforms. Langfuse, LangSmith, Arize AX, and Helicone monitor LLM pipelines in production: tracing, token costs, latency, prompt regressions. Langfuse[3] stands out with its open-source codebase (MIT license) and self-hosting capability; LangSmith—with the deepest LangChain integration; Arize AX—with the most mature evaluation primitives (embedding drift, anomaly detection).

Ecosystem 4: Off-the-shelf conversation analytics and QA platforms. *QA platforms embedded in CX ecosystems:* Zendesk QA (formerly Klaus)[4] automatically scores 100% of conversations with churn, loop, and escalation detection—but operates exclusively within Zendesk. Microsoft Dynamics 365 Quality Scoring[5] (announced in Release Wave 1, 2026) introduces AI-driven quality assessment with the Quality Evaluation Agent. Amazon Connect Contact Lens[6] uses deep learning for transcription and sentiment change identification. *Conversation intelligence platforms (sales-focused):* Gong, Chorus (ZoomInfo), Clari, and Revenue.io analyze sales conversations: they identify breakthrough moments, measure talk ratio, detect objections, and score sales opportunities. They target a different domain (sales pipeline, not customer service). *Contact center QA platforms:* Observe.AI[7], MaestroQA, Solidroad, and AmplifAI score agent interactions against defined rubrics. They are close to the proposed framework in ambition but evaluate human agents, not bots, and do not offer a taxonomy grounded in service recovery literature. *Customer feedback / conversation analytics platforms:* Echo AI, Zonka Feedback, BuildBetter, Nextiva, and Intercom (CX Score) analyze transcripts for topics, sentiment, and CX trends at the aggregation level, not at the level of individual sessions with detection of specific relational risks.

[2] `https://docs.confident-ai.com/` [accessed: July 2026]
[3] `https://langfuse.com/docs` [accessed: July 2026]
[4] `https://www.zendesk.com/service/quality-assurance/` [accessed: July 2026]
[5] `https://learn.microsoft.com/en-us/dynamics365/customer-service/administer/quality-management` [accessed: July 2026]
[6] `https://docs.aws.amazon.com/connect/latest/adminguide/contact-lens.html` [accessed: July 2026]
[7] `https://www.observe.ai/product` [accessed: July 2026]