Mind the Goal: Data-Efficient Goal-Oriented Evaluation of Conversational Agents and Chatbots using Teacher Models

Deepak Babu Piskala   **Sharlene Chen**   **Udita Patel**
Parul Kalra   **Rafael Castrillo**
Amazon.com, Seattle, WA, USA

Abstract

Evaluating the quality of multi-turn chatbot interactions remains challenging, as most existing methods assess interactions at the turn level without addressing whether a user's overarching goal was fulfilled. A "goal" here refers to an information need or task, such as asking for policy information or applying for leave. We propose a comprehensive framework for goal-oriented evaluation of multi-agent systems (MAS), introducing the **Goal Success Rate (GSR)** to measure the percentage of fulfilled goals, and a **Root Cause of Failure (RCOF)** taxonomy to identify reasons for failure in multi-agent chatbots. Our method segments conversations by user goals and evaluates success using all relevant turns. We present a model-based evaluation system combining teacher LLMs, where domain experts define goals, set quality standards serving as a guidance for the LLMs. The LLMs use "thinking tokens" to produce interpretable rationales, enabling *explainable*, *data-efficient* evaluations. In an enterprise setting, we apply our framework to evaluate AIDA, a zero-to-one employee conversational agent system built as a ground-up multi-agent conversational agent, and observe GSR improvement from 63% to 79% over six months since its inception. Our framework is generic and offers actionable insights through a detailed defect taxonomy based on analysis of failure points in multi-agent chatbots, diagnosing overall success, identifying key failure modes, and informing system improvements.

1 Introduction

Modern conversational assistants increasingly adopt agentic LLM architectures, in which a central reasoning model (DeepSeek-AI et al., 2025; Anthropic, 2025; OpenAI, 2025) coordinates tool invocation, external-memory reads/writes, and multi-step planning to accomplish user tasks (Liu et al., 2025; Jacovi and Goldberg, 2023). Such agents learn to call APIs, query databases, or operate
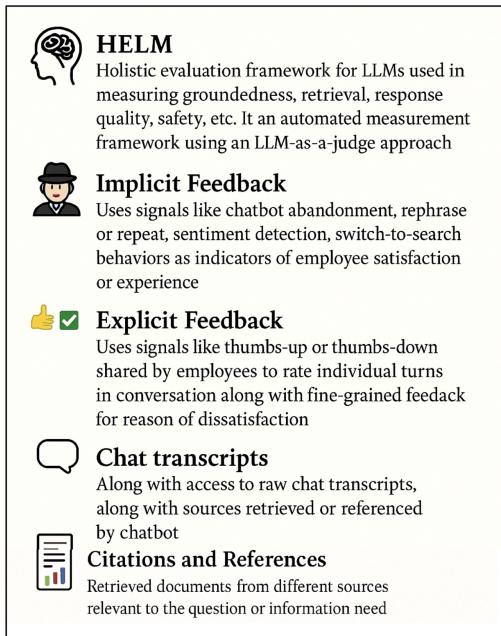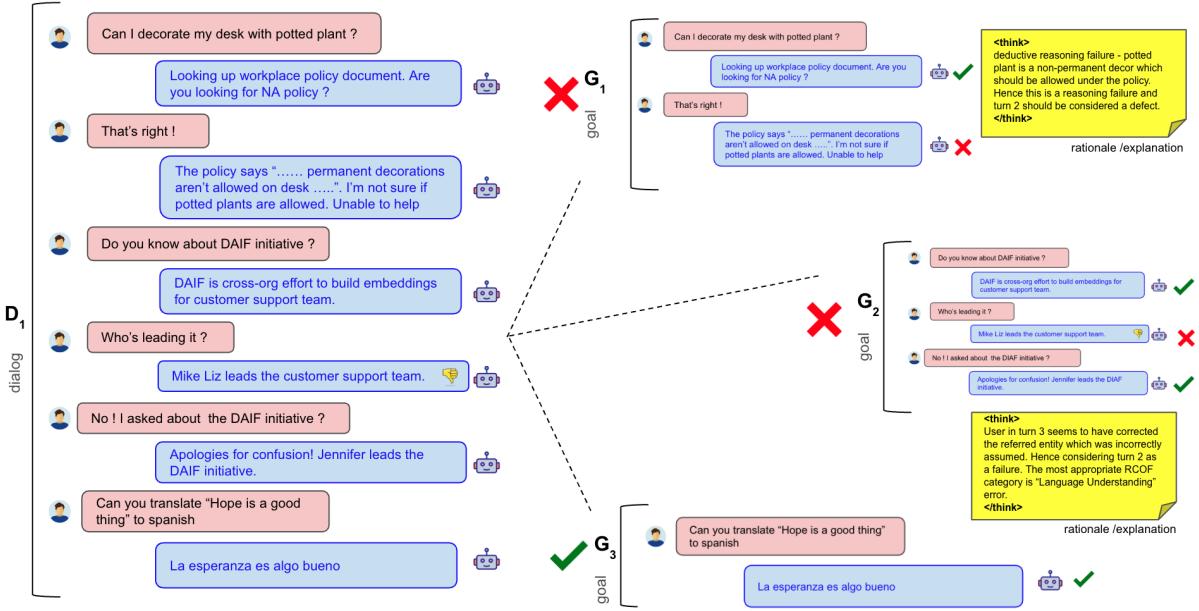

Figure 1: Signals used for evaluating chatbot quality

forms—often teaching themselves new tool affordances in the wild (Schick et al., 2023; Qin et al., 2024). To support long-horizon dialogues and personalization, they maintain dedicated episodic or vector memories that can be dynamically written and retrieved (Salama et al., 2025; Du et al., 2023). Sophisticated reasoning routines (e.g., chain-of-thought or program-of-thought prompts) enable the agent to decompose queries, condition tool use on intermediate results, and verify answers before replying (Wei et al., 2022; Chen et al., 2022). At enterprise scale, multiple specialized agents—HR, IT, legal, analytics—often collaborate through emerging interoperability protocols such as Anthropic's Model Context Protocol (MCP) and Google's Agent-to-Agent (A2A) standard, forming loosely coupled multi-agent systems (Hou et al., 2025; Khandelwal et al., 2025). While this ecosystem unlocks powerful organizational workflows,

Figure 2: **Goal-oriented breakdown of a multi-turn employee chatbot conversation.** The full dialog (left) is segmented into three distinct goals: G_1 (policy inquiry), G_2 (project clarification), and G_3 (translation request). Each goal is independently evaluated for success or failure using goal-level metrics. Turn-level evaluation may suggest high success, but goal-level evaluation reveals that G_1 and G_2 failed due to reasoning and language understanding errors, respectively. Root Cause of Failure (RCOF) is annotated using structured rationale snippets (right), highlighting the earliest defective turn per goal.

every new memory layer, tool wrapper, or inter-agent message channel compounds the risk of subtle cascading failures, underscoring the need for robust, interpretable evaluation.

Chatbots and conversational assistants are increasingly used to handle complex information-seeking and action-taking dialogues. Ensuring high-quality interactions is critical, especially as users may ask follow-up questions or rephrase queries until their goal is met. However, evaluating chatbot quality in a meaningful way remains challenging. Today, most chatbot evaluations focus on individual turn-level metrics (each user query and the bot's response), such as response relevance (Patel et al., 2025). While these metrics provide useful signals, they often fall short of capturing the full picture of user satisfaction — in particular, whether the user's *underlying goal* was eventually achieved across the entire conversation.

There is a need for a more holistic evaluation framework that moves beyond isolated turns to assess the success of a conversation as a whole. Current systems lack a unified view to diagnose end-to-end conversational success or identify where in a multi-turn exchange the assistant failed to meet the user's needs. For example, an enterprise chatbot

may involve components for retrieval, language understanding, and external tool or database calls; a failure in any one component can cause the conversation to derail.

In practice, multi-turn dialogs are common and particularly prone to failures. In our analysis of an enterprise conversational assistant called AIDA, we found that about 39% of dialogs involve multiple turns, and these multi-turn dialogs contribute to a disproportionate share of user frustration. For instance, multi-turn sessions exhibited a negative feedback rate (e.g. user explicitly indicating dissatisfaction) roughly three times higher than single-turn sessions (2.65% vs 0.9%).

We propose a goal-oriented evaluation framework for chatbots that segments each dialog into user-defined **goals** and measures a strict *Goal Success Rate (GSR)* — a goal is marked successful only if *all* its turns are error-free. To make this metric actionable, we also introduce a *Root Cause of Failure (RCOF)* taxonomy that attributes each failed goal to a predefined error category, enabling developers to pinpoint dominant failure modes. In summary, our contributions include:

- A general **goal-oriented evaluation frame-**

work for multi-turn conversations, which segments dialogs by user goals and evaluates success at the goal level.

- Definition of the **Goal Success Rate (GSR)** metric to quantify the fraction of user goals that are satisfied, and a **Root Cause of Failure (RCOF)** taxonomy to categorize and explain failed goals.

- A **model-based implementation** using a large language model as a "teacher" to assist in labeling goals and turn outcomes, demonstrating how GSR and RCOF can be computed on real chatbot logs.

- Empirical analysis on enterprise chatbot conversations, showing that the framework captures holistic quality signals (e.g. lower GSR for multi-turn queries, identification of top failure reasons) and discussing how these insights drive system improvements.

2 Related Work

Evaluating open-domain and task-oriented dialog systems has been an active research area. Traditional evaluation metrics for chatbots often operate at the *utterance* level. For example, automated metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) compare generated responses to reference texts, while embedding-based scores like BERTScore (Zhang et al., 2020) aim to capture deeper semantic similarity. More recently, large language models themselves have been employed as judges—rating responses against conversational guidelines or rubrics (e.g. G-Eval (Liu et al., 2023) and MT-Bench evaluations (Zheng et al., 2023)). However, these turn-level metrics do not directly indicate whether the user's *broader goal* was accomplished.

In task-oriented dialogue systems, it is common to measure *task success* or *goal completion rate* (Lu et al., 2020a). These metrics typically require a predefined goal (such as booking a restaurant or answering a specific question) and often rely on comparing the dialog outcome to a known target or using user self-reports. Our work is inspired by task success metrics (Lu et al., 2020a) but extends the idea to more open-ended, user-driven conversations where the goals must be inferred and may evolve.

Another line of related work focuses on user satisfaction (Fu et al., 2022) and engagement as metrics for dialog quality. For example, systems have been evaluated based on user ratings, re-engagement rates, or negative feedback signals (e.g. explicit thumbs-down clicks) (Hancock et al., 2019; Mehri and Eskenazi, 2020). These signals provide valuable supervision for quality, and our framework could integrate them (e.g. as features or validation for goal success labels). Nonetheless, they are often sparse and do not give detailed reasons for failure.

In terms of understanding failure modes, prior work in error analysis of conversational systems and virtual assistants has introduced taxonomies of errors (for instance, distinguishing between interpretation errors vs. knowledge retrieval errors) (Ram et al., 2018). Our RCOF taxonomy is aligned with these ideas, but tailored to the enterprise chatbot scenario and designed to be used in an automated evaluation pipeline. Lastly, we leverage large pre-trained language models to assist evaluation, which relates to the growing trend of using AI models as proxy evaluators or "rubric generators" for AI outputs (Zhang et al., 2022; Mao et al., 2024). Our use of a teacher model demonstrates how such models can help segment and label conversation quality at scale, while a human-in-the-loop ensures the reliability of these labels for building goal-level metrics. Lu et al. (Lu et al., 2020b) propose a BERT-based span prediction model to optimize the identification of relevant contextual utterances in task-oriented dialogues. This approach enhances the calculation of Goal Success Rate (GSR) by accurately segmenting user goals, thereby reducing labeling waste and potential bias in evaluation metrics.

3 Anatomy of a Conversation

To effectively evaluate chatbot interactions, we begin by defining the fundamental units of analysis: **session**, **goal**, and **turn**. These definitions allow us to formally model dialogue structure and success.

- A **session** (S) refers to a full interaction between a user and the chatbot, typically bounded by a timeout or user exit (also called dialog).

- Each session consists of one or more **goals** (G_i), where a goal represents a coherent user intent or information need (e.g., "Where can I submit expenses?").

- A **goal** is realized as a sequence of one or more **turns** (T_j), with each turn comprising a user query q_j and the chatbot response r_j.

A goal is marked *successful* if all its turns are successful[1]—i.e., the bot provided correct and helpful responses. If any turn within a goal fails, the entire goal is considered failed. This strict criteria ensures high fidelity to user experience.

Figure 3 illustrates an example session S_1 comprising three goals (G_1, G_2, and G_3), each containing one or more turns. The first two goals are successful, while the third fails due to missing knowledge, prompting the user to abandon the chat.
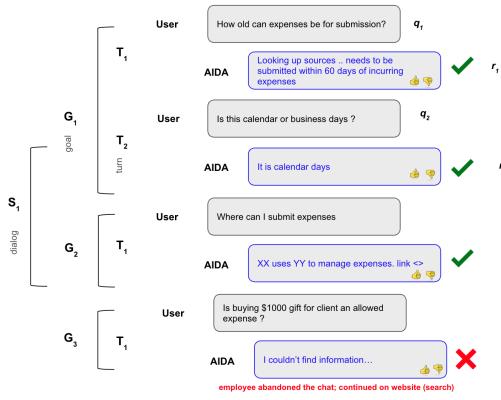

Figure 3: Anatomy of a chatbot session: Each dialog consists of multiple goals (G_i), where each goal comprises one or more turns (T_j), formed by a query-response pair (q_j, r_j). Goal G_3 is marked failed due to information unavailability.

4 Framework for Goal-Oriented Evaluation

Our framework, which we refer to as a Conversational Intelligence Model (CIM) for chatbot evaluation, consists of three main components: (1) **Goal Segmentation**, (2) **Goal Success evaluation (GSR)**, and (3) **Root Cause of Failure (RCOF) attribution**. We describe each in turn.

4.1 Goal Segmentation

A conversational session (dialog) is a sequence of turns, where each turn typically consists of a user's message (question) and the chatbot's response. In goal segmentation, the aim is to identify boundaries between distinct *goals* within a dialog. Intuitively, a new goal begins when the user asks something that constitutes a new or different information need,

as opposed to continuing or clarifying the previous question.

In formal terms, for each turn T_j in a dialog (for $j = 1, 2, \ldots, N$), we predict a label *is_new_goal*(T_j) $\in \{yes, no\}$ indicating whether T_j starts a new goal. The first turn of a dialog is always a new goal by definition. A label of *yes* means the user's utterance in turn T_j is considered the start of a new goal (information need), whereas *no* means the turn is continuing the previous goal. By applying this segmentation, a single dialog of N turns can be divided into one or more goals $G_1, G_2, \ldots, G_K$, where K is the number of turns labeled as starting a new goal. Each goal G_k consists of a contiguous sequence of turns addressing a particular query or task.

Goal segmentation can be seen as a classification task at the turn level. Features for this task can include lexical cues (e.g. the user asks an unrelated question indicating a topic switch), contextual cues (e.g. the user explicitly says "Now I have another question..."), or even temporal gaps (if a conversation resumes after a long pause, it might indicate a new goal). In our implementation, we utilize a large language model to examine the conversation and predict these boundaries, as described in 6. Accurate segmentation is important, since it directly affects the granularity at which success is measured.

4.2 Goal Success Rate (GSR)

Once goals are identified in a conversation, we evaluate the quality of each goal by examining the turns within it. We define a **goal** as *successful* if **every turn in that goal is successful**. Conversely, if any turn in the goal was a failure (e.g. the chatbot's answer was incorrect, irrelevant, or otherwise unsatisfactory for that turn), then the entire goal is labeled as *failed*. This definition is deliberately strict: even if the user eventually gets the answer after rephrasing their question in a later turn, we consider the goal to have failed because the user had to work through a failed response along the way. In other words, the assistant must get it right on the first attempt *and* continue to be correct for all follow-ups to count as a success under this metric. This high standard ensures that our evaluation emphasizes truly seamless interactions.

Formally, let each goal G_k consist of turns $T_{s_k}, T_{s_k+1}, \ldots, T_{e_k}$, where T_{s_k} is the first turn of goal k and T_{e_k} is the last turn (right before the next goal starts or the dialog ends). We have a func-
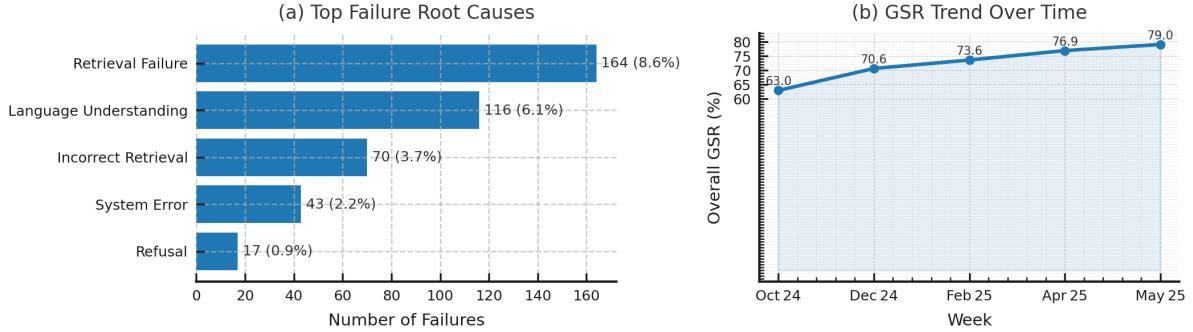
---
[1]Alternate formulations include assigning fractional credit based on the number of successful turns, or defining goal success based on the final turn alone. We discuss these in the Future Directions section.

Figure 4: **(a)** Distribution of the top five chatbot failure root causes, annotated with both absolute counts and percentages. **(b)** Trend of the overall Goal-Success Rate (GSR) from Oct '24 through May '25, showing a steady improvement.

tion $quality(T_j)$ which outputs *success* or *failure* for each turn (how to determine turn success is discussed later). Then the quality of goal G_k is:

$$
GoalQuality(G_k) = \begin{cases} success, & \text{if } \forall\, T_j \in G_k,\ quality(T_j) = success, \\ failure, & \text{otherwise.} \end{cases}
$$

The **Goal Success Rate (GSR)** is defined as the fraction of goals in a dataset that are successful. If we have K goals in total (across some set of dialogs), and we use an indicator function $1[\cdot]$ that is 1 if a goal is successful and 0 if failed, then:

$$
\text{GSR} = \frac{1}{K}\sum_{k=1}^{K} 1[goalQuality(G_k) = success] \times 100\%
$$

4.3 Root Cause of Failure (RCOF) Taxonomy

While GSR provides a high-level success metric, it does not explain *why* goals fail. To enable actionable insights, we introduce a taxonomy of seven error categories—termed **Root Cause of Failure (RCOF)**—each corresponding to a distinct breakdown in chatbot behavior: **E1** Language Understanding Failure, **E2** Refusal to Answer, **E3** Incorrect Retrieval, **E4** Retrieval Failure, **E5** System Error, **E6** Incorrect Routing, and **E7** Out-of-Domain or Unsupported Query. Each failed turn is annotated with one of these codes, and we assign a goal's RCOF based on the earliest failed turn, under the assumption that initial breakdowns are most disruptive. This framing helps prioritize debugging and aligns naturally with engineering signals (e.g., E3 with low coverage, E5 with timeouts), making the taxonomy interpretable and actionable. A

detailed description of each RCOF category is provided in **Appendix B**.

5 Data

We evaluate our GSR framework on real-world interaction logs from **AIDA**, an enterprise-grade virtual assistant deployed across desktop and mobile to help employees with workplace queries spanning HR, IT, wiki, expenses, and internal tools. The dataset comprises approximately ~10,000 multi-turn conversations collected over 30 days, where each session may embed multiple user **goals**—ranging from informational queries to action-oriented tasks like leave applications or meeting room bookings. Each turn is annotated with rich **implicit signals** (rephrases, abandonments, search fallback), **explicit feedback** (likes, thumbs up/down), and **metadata** such as device type, timestamp, and retrieved citations. A detailed breakdown of dataset composition and feedback signals is provided in **Appendix A**.

6 Methodology

To generate ground-truth annotations for evaluating chatbot quality, we adopt a human-in-the-loop (HITL) pipeline that combines expert-defined SOPs with LLM-based multi-teacher supervision. The methodology applies across all stages of evaluation: goal segmentation, success classification (GSR), and root cause attribution (RCOF).

Figure 5 illustrates our end-to-end setup. We begin with normalized event logs collected from AIDA, an enterprise chatbot. These logs are preprocessed into a linked dialog dataset, grouping message turns into coherent multi-turn conversa-

tions. We oversample longer sessions to ensure coverage of complex goals.

We then sample N conversations and evaluate each using a set of three independently prompted foundation models (FMs), referred to as expert teacher models—examples include Claude Sonnet, Claude Haiku (Anthropic, 2024), GPT-4 (OpenAI, 2023), and LLaMA-4 (AI, 2025). Each model is invoked using Chain-of-Thought (CoT) prompting, wherein the model is instructed to use explicit reasoning tags ('<think> ... </think>') before outputting its final quality judgment or label. This promotes reflective system-2 style thinking and enables richer, interpretable rationales.

Once each expert provides its opinion on a given goal (e.g., whether it was fulfilled or which failure label applies), we aggregate the responses via majority voting. If two or more teacher models agree on a label, we accept the annotation as ground truth. In cases where all three models disagree, we mark the goal as *ambiguous* and escalate to human annotators. These experts refer to business-specific SOPs to resolve such edge cases and refine definitions of quality. Over time, this feedback loop refines model behavior and improves inter-model consistency.

Once the labeled goal dataset is produced, we optionally distill the teacher ensemble into a lightweight student model for efficient real-time and offline inference. The student, trained on the voted labels, mimics teacher decisions at a lower cost.
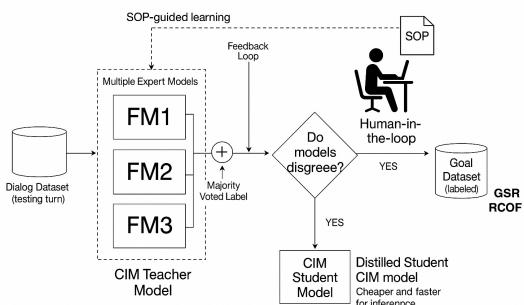

Figure 5: HITL evaluation pipeline: AIDA conversations are processed by multiple expert models using Chain-of-Thought prompts. Majority-voted outputs are accepted as labels. Disagreements are escalated to human experts guided by SOPs.

7 Results

We evaluated the GSR framework on a stratified sample of approximately 10,000 dialogs from **AIDA**, as described in Section 5. The overall Goal Success Rate (GSR) was 78%, indicating that a majority of user goals were successfully resolved end-to-end without any defective turns. However, the GSR dropped to 66% for multi-turn goals (those spanning two or more turns), revealing that conversational complexity significantly increases the risk of failure. A breakdown of goal failures using our RCOF taxonomy is shown in Table 1, where the top three failure types were retrieval failures (39%), language understanding errors (27%), and incorrect retrievals (16%). This confirms that while AIDA handles single-turn factoid queries well, it is more prone to error in nuanced or multi-step interactions where accurate retrieval and contextual comprehension are critical.

Metric	Count	% of Goals
Total Goals (sample)	1915	100%
Successful Goals	1488	77.7%
Failed Goals	427	22.3%
Top failure root causes:		
Retrieval Failure (E4)	164	8.6%
Language Understanding (E1)	116	6.1%
Incorrect Retrieval (E3)	70	3.7%
System Error (E5)	43	2.2%
Refusal (E2)	17	0.9%

Table 1: Goal Success Rate and failure breakdown for a sample of AIDA chatbot dialogs. The top section shows overall GSR, and the bottom lists top root causes of failed goals.

Over a three-month period, AIDA evolved from a basic retrieval-augmented system to an agentic LLM-powered assistant capable of reasoning, invoking tools, and managing contextual queries. As shown in Figure 4, we observed a steady improvement in goal completion rates—from 64% in February to 78% by April. This growth was not driven by prompt tuning or fallback rules, but rather by launching new capabilities: improved source integration, routing mechanisms, upgraded models with better language reasoning, and more flexible agentic behaviors (e.g., issuing clarification questions or synthesizing multi-source answers). Notably, the GSR for multi-turn goals rose by 12 points, demonstrating the practical utility of our evaluation framework in guiding and validating iterative system improvements. We assess teacher

model reliability by comparing its labels against expert human annotations; detailed agreement statistics and analysis are provided in **Appendix C**.

Limitations

While our proposed goal-oriented evaluation framework offers a structured and scalable way to assess chatbot quality, it has certain limitations.

First, our evaluation methodology is best suited for task-oriented and information-seeking dialogs with clear success criteria. In open-ended scenarios such as summarizing documents or composing emails, quality becomes highly subjective and user-dependent. Behavioral signals like thumbs-downs, chat termination, or repeated clarifications can offer implicit supervision, but such signals are sparse and often unavailable in real-world logs.

Second, we assume that each goal corresponds to a contiguous sequence of turns. However, in complex or compound dialogs, user goals may span non-consecutive turns or interleave with others (e.g., returning to a prior topic). Our current segmentation method does not support such dependencies. Future work could explore modeling dialog goals as graph structures to capture cross-references, co-references, and interleaved subgoals more accurately.

Third, our current evaluation may understate hallucinations—cases where the assistant generates fluent but factually incorrect information. Without external verification or user behavior signals to flag discrepancies, such errors may go undetected. Addressing hallucination detection remains a broader challenge, especially in enterprise scenarios where accurate grounding in internal knowledge bases is critical.

References

Meta AI. 2025. LLaMA – 4: Next-generation open foundation model. https://ai.meta.com/blog/llama-4/. Accessed 14 May 2025.

Anthropic. 2024. Claude 3 Haiku: Lightweight frontier LLM. https://www.anthropic.com/news/claude-3-model-family. Accessed 14 May 2025.

Anthropic. 2025. Claude 3.7 sonnet and claude code. https://www.anthropic.com/news/claude-3-7-sonnet. Product card describing the reasoning capabilities of Claude 3.7 Sonnet.

Andy T. Chen, Welezha Zhang, Barun Patra, and Mohit Bansal. 2022. Program-of-thoughts prompting: Disentangling logic from surface form. *arXiv preprint arXiv:2211.12588*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, and Ruoyu Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhengxuan Du, Weize Chen, Zhewei Yao, Eric P. Xing, and Yuandong Tian. 2023. Longmem: Scaling language models with long-term memory. *arXiv preprint arXiv:2307.14995*.

Tingchen Fu, Shen Gao, Xueliang Zhao, Ji rong Wen, and Rui Yan. 2022. Learning towards conversational ai: A survey. *AI Open*, 3:14–28.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazaré, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *Preprint*, arXiv:1901.05415.

Xing Hou, Yingxuan Yang, Huacan Chai, Yuanyi Song, and Siyuan Qi. 2025. A survey of agent interoperability protocols: Model context protocol, agent cards and beyond. *arXiv preprint arXiv:2505.02279*.

Aviad Jacovi and Yoav Goldberg. 2023. Coala: Cognitive agents for language applications. *arXiv preprint arXiv:2309.02427*.

Renu Khandelwal, Mert Cemri, and Melissa Pan. 2025. Building a secure agentic ai application leveraging the agent-to-agent (a2a) protocol. *arXiv preprint arXiv:2504.16902*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proc. of the ACL Workshop*, pages 74–81.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, and Sirui Hong. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2511–2522.

Weiyi Lu, Yi Xu, and Erran Li. 2020a. Efficient evaluation of task oriented dialogue systems. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.

Weiyi Lu, Yi Xu, and Erran Li. 2020b. Efficient evaluation of task-oriented dialogue systems. In *NeurIPS 2020 Workshop on Human in the Loop Dialogue Systems*.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2024. GPTEval: A survey on assessments of ChatGPT and GPT-4. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, page 5754–5765.

Shikib Mehri and Maxine Eskenazi. 2020. Usr: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 681–707.

OpenAI. 2023. Gpt-4 technical report. https://cdn.openai.com/papers/gpt-4.pdf. Accessed 14 May 2025.

OpenAI. 2025. Introducing openai *o3* and *o4-mini*. https://openai.com/index/introducing-o3-and-o4-mini/. Company blog post announcing the *o3* reasoning model.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Udita Patel, Rutu Mulkar, Jay Roberts, Cibi Chakravarthy Senthilkumar, Sujay Gandhi, Xiaofei Zheng, Naumaan Nayyar, Parul Kalra, and Rafael Castrillo. 2025. Thelma: Task based holistic evaluation of large language model applications-rag question answering. *Preprint*, arXiv:2505.11626.

Liuqing Qin, Yihong Chen, Weijian Li, Zhangyin Feng, and Yong Jiang. 2024. A survey on tool learning with foundation models. *arXiv preprint arXiv:2405.17935*.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2018. Conversational ai: The science behind the alexa prize. *Communications of the ACM*.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760*.

Timo Schick, Jonas Bär, Hendrik Schütze, and Hinrich Schütze. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04782*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, and Brian Ichter. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Chen Zhang, Luis Fernando D'Haro, Qiquan Zhang, Thomas Friedrichs, and Haizhou Li. 2022. Fined-eval: Fine-grained automatic dialogue-level evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 3332–3347.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Appendix

A   Dataset Composition

We evaluate our GSR framework using real-world interaction logs from an enterprise-grade chatbot, called AIDA deployed to assist employees with workplace-related queries. The chatbot serves as a virtual assistant capable of addressing a wide variety of topics including HR policies, IT troubleshooting, expense reimbursement, time-off requests, and access to internal tools or documentation. The assistant operates in natural language and is available via chat platforms on desktop and mobile, used internally by the organization. In addition to answering informational queries, AIDA can handle action-oriented goals such as applying for sick leave on behalf of an employee or booking a meeting room.

Our dataset comprises ∼10,000 multi-turn conversations collected over a 30-day window. Each conversation represents a **session**, which may contain multiple **goals**, where a goal corresponds to a specific user intent or task (e.g., checking leave balance, updating benefits, submitting expenses). The data includes:

- **User utterances and bot responses**: Complete conversational transcripts segmented into turns.

- **Implicit feedback signals**: Indicators such as query rephrasing, abandonment, switches to search, and delayed user responses.

- **Explicit feedback**: Thumbs up/down on responses, likes, and internal reshares.

- **Citations and References**: Includes RAG articles used for answering the query, tool outputs, etc. along with metadata of conversationa like time, device(mobile/desktop) and user attributes.

B   Detailed RCOF Definitions

While GSR provides a single-number summary of success, it does not explain *why* goals failed. For actionable insights, we need to analyze the failures in more detail. We introduce a predefined taxonomy of error categories to attribute each failed goal to a *root cause of failure (RCOF)*. Each failure category corresponds to a general type of breakdown that can occur in a chatbot's handling of a query.

Drawing from common issues in information-seeking and action-taking dialogues, we define seven distinct root cause categories (which we label E1 through E7 for convenience):

- **Language Understanding Failure (E1)**: The assistant misunderstood the user's request or context, leading to an irrelevant or incorrect answer. For example, the user says "cancel my request," and the bot misinterprets "cancel" in the wrong context.

- **Refusal to Answer (E2)**: The assistant inappropriately refused to answer the question (or gave a safe completion) even though it should have been able to help. In other words, no disallowed content was present, but the bot still responded with a refusal.

- **Incorrect Retrieval (E3)**: The assistant retrieved the wrong informational content. This is specific to systems that use a retrieval-augmented generation (RAG) or knowledge base: the bot did fetch some documents or data, but those turned out to not contain the answer needed (so the answer was inevitably wrong or incomplete).

- **Retrieval Failure (E4)**: The assistant failed to retrieve any relevant information when it should have. For instance, the user asked a factual question answerable from a knowledge base, but the system returned no results (perhaps due to a search/query failure).

- **System Error (E5)**: A technical issue prevented a correct answer. This could include the response getting cut off (e.g. a timeout or the generation stopping mid-sentence) or an integration failure. Essentially, the system could not produce a proper answer due to an error or glitch.

- **Incorrect Routing (E6)**: The user's query was routed to the wrong domain or module of the assistant. In enterprise assistants that orchestrate multiple bots or skill routes, a question might be answered by an inappropriate knowledge category (for example, a question meant for an HR database was mistakenly handled by a general FAQ bot), leading to a faulty answer.

- **Out-of-Domain or Unsupported Query (E7)**: The user's request is outside the scope

of what the assistant is designed to handle (e.g. asking a legal question to an IT support bot), or involves capabilities it doesn't have (like requesting a translation if that's not supported). In such cases, failure is expected because the question is invalid for the system.

Each failed turn in a conversation is annotated with an RCOF code (E1–E7) to indicate the failure type. Since a single goal may contain multiple failed turns, we define the **root cause of a failed goal** as the error category of its *earliest failed turn*. This heuristic assumes that the initial breakdown is the most influential in derailing the goal and helps focus analysis on the first error rather than compounding effects. For example, if the assistant misunderstood the question (E1) early on, we attribute the goal's failure to language understanding—even if a retrieval failure occurred later. In other words, within a failed goal G_k, let T_j be the first turn (in chronological order) marked as failure; the RCOF label assigned to T_j becomes the root cause for G_k. This approach guides debugging toward root issues rather than symptoms. Moreover, RCOF categories can be aligned with internal system metrics (e.g., low source coverage may indicate E3, or frequent fallbacks may reflect E5), making the taxonomy both interpretable and actionable for engineering teams.

C Human–LLM Agreement

Agreement with Human Annotators: To assess the reliability of our teacher model ensemble, we conducted a comparison against expert human annotators using the same SOP guidelines provided to the LLMs. We found that human reviewers agreed with model-generated labels in approximately **75%** of cases. In the remaining **25%**, there was at least one point of disagreement across the three annotation tasks: *goal segmentation*, *turn-level quality*, and *root cause attribution (RCOF)*.

When analyzing task-specific agreement, we observed that only **13%** of dialogs showed disagreement between humans and LLMs for either goal segmentation or turn quality evaluation. However, for RCOF attribution, disagreement rose to **17%** of cases. We attribute these gaps to *ambiguity in the SOP definitions*, where both humans and models encountered unclear guidance for edge cases or subjective interpretations. We expect to reduce such discrepancies to below **5%** through *closed-loop*

train–evaluate cycle that integrates human feedback into the teacher model prompting strategy.

D LLM Prompt Template

```
system_prompt = "You are a helpful AI assistant.
    You will act as a judge to evaluate quality
    of employee experience chatbot."

output_format = """
{
  dialog_id: xx,
  turns: [
    {turn_number: 1, is_new_goal: yes/no, quality
        : success/failure, rcof: E1-E7 | null},
    {turn_number: 2, is_new_goal: yes/no, quality
        : success/failure, rcof: E1-E7 | null},
    ...
  ]
}

where
  is_new_goal \in {yes,no} # compare adjacent
      user turns
  quality \in {success,failure} # based on
      response + follow-ups
  rcof \in {E1-E7} if failure else null

RCOF codes
  E1 Incorrect Sources - irrelevant docs
      retrieved
  E2 Retrieval Failure - no docs retrieved
  E3 Refusal to Answer - unwarranted refusal
  E4 Language Understanding - misinterprets
      question
  E5 System Error - blank / truncated response
  E6 Incorrect Routing - wrong domain/department
  E7 Out-of-Domain Query - capability not
      supported
"""

template = """
{system_prompt}
You are provided with a dialog from an employee
    chatbot.
Output the JSON for every turn, reasoning inside
    <think>...</think> tags
but printing *only* the JSON.

output format:
{output_format}

input:
{question}
"""
```

E Custom JSON Schema

Schema overview. We store every annotated dialog as a JSON object with a unique `dialog_id` and an array of per-turn records. Listing 1 shows the structure (placeholders <...> indicate value types).

Listing 1: Abstract JSON schema for dialog annotations used in our pipeline

```
{
  "dialog_id": <string>, // UUID for the dialog
  "turns": [
    {
      "turn_number": <int>,
      "user_msg": <string>,
      "response": <string>,
      "source_urls": <string[]>,
      "source_names": <string[]>,
      "source_snippets": <string[]>
    },
    ...
  ]
}
```

Field descriptions.

- **dialog_id** – Globally unique identifier (UUID v4) used to join logs and annotations.

- **turn_number** – Sequential index starting at 1; enables mapping annotations back to raw logs.

- **user_msg** / **response** – Raw text of the employee's utterance and the chatbot's reply.

- **source_urls** – List of URLs or internal document IDs returned by the retrieval component.

- **source_names** – Optional human-friendly titles corresponding to each URL (may be empty).

- **source_snippets** – Evidence snippets ($\leq$256 chars each) extracted from the retrieved sources and shown to the user.

This schema underpins both the teacher-model annotation pipeline and downstream analytics, ensuring that every evaluation label can be traced back to its conversational and knowledge context.