Learning from Corrections: A Continual Learning Architecture for Conversational AI with Real-Time User Feedback

Prof. Daniel Kim

[1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, United States

e-mail: dkim@mit.edu.

ABSTRACT Conversational AI systems have become integral across domains, yet most remain static after deployment, unable to learn from ongoing user interactions. This limitation leads to repeated errors, reduced personalization, and diminished user satisfaction. This paper proposes a continual learning architecture that enables dialogue models to improve in real time by incorporating user corrections—both explicit and implicit—as feedback signals. The framework integrates a feedback listener, episodic memory buffer, and hybrid continual learning engine combining Elastic Weight Consolidation (EWC) and Experience Replay. These modules ensure adaptability while mitigating catastrophic forgetting, maintaining a balance between stability and plasticity. Experiments conducted with synthetic and real feedback datasets demonstrate significant performance gains: the Correction Success Rate increased to 87%, catastrophic forgetting was contained below 5%, and linguistic accuracy improved (BLEU score from 23.5 to 29.1). Although modest overheads in latency and memory usage were observed, the system sustained deployment feasibility in real-world contexts. The findings highlight the potential of real-time feedback-driven continual learning to transform conversational AI into adaptive, evolving agents capable of personalization, domain adaptation, and long-term reliability across critical sectors such as healthcare, education, and enterprise services.

I. INTRODUCTION

Conversational AI has witnessed a rapid rise in adoption across diverse domains, from personal voice assistants like Siri and Alexa to enterprise-level chatbots and advanced dialogue models such as ChatGPT. These systems are typically built on large-scale pre-trained language models capable of generating fluent and contextually relevant responses. However, one critical limitation remains: these systems are often deployed as static entities, unable to learn or adapt from new user interactions after deployment. Consequently, they are prone to repeating the same mistakes, misinterpreting user intents, or failing to update based on evolving dialogue contexts, which significantly impairs user engagement and satisfaction [1], [2].

Traditional models rely on offline training pipelines, where improvements are made retrospectively through periodic updates based on manually curated datasets. In real-world usage, however, users often provide valuable correctional input—for example, “No, I meant tomorrow”—but the system lacks the architecture to learn from such feedback dynamically. This disconnect between user interaction and model learning leads to inefficiencies and frustrates users who are forced to rephrase or reissue their queries [3]. Moreover, this inability to incorporate live feedback also restricts personalization, as the agent treats each user interaction in isolation, with no memory or evolving understanding of user preferences.

Incorporating real-time user feedback into the learning loop presents a promising pathway for creating adaptive and intelligent conversational agents. Corrections, whether explicit or implicit, can serve as meaningful supervision signals that guide the model to avoid repeating errors and to better understand nuanced intent. However, integrating such continual learning in conversational AI brings technical challenges. A major concern is catastrophic forgetting, wherein learning from new inputs may overwrite prior knowledge, degrading performance on earlier tasks or dialogues [4]. Addressing this requires balancing the stability–plasticity dilemma—the need to retain old knowledge while remaining flexible enough to learn new patterns [5], [6].

Although some systems, such as Meta’s BlenderBot 3 and Amazon’s Alexa Teacher Model, have explored human-in-the-loop mechanisms, these typically rely on batch-learning updates rather than real-time continual learning [7]. The broader application of continual learning in language systems remains underexplored, particularly in settings involving noisy or ambiguous user feedback and tight latency constraints.

The significance of this study lies in its potential to reshape how conversational systems interact with users—not merely as static responders, but as evolving, context-aware agents. By enabling real-time learning from corrections, this research enhances personalization, increases response accuracy, and reduces the cognitive load on users. Furthermore, such adaptive systems could prove invaluable in high-stakes domains like healthcare triage, legal assistance, or crisis management, where the cost of misunderstanding user intent is high and rapid learning from dialogue history is essential.

A. AIM

To design and evaluate a continual learning architecture for conversational agents that improves over time by learning from user corrections in real time.

B. OBJECTIVES:

1) To capture and categorize real-time user feedback (explicit and implicit) as learning signals.
2) To integrate continual learning mechanisms (e.g., replay, regularization) into dialogue model pipelines.
3) To assess model adaptability, user satisfaction, and error recurrence post-feedback.
4) To analyze trade-offs between performance gains, latency overhead, and resource demands.

This paper proceeds as follows. Section 2 reviews the literature on conversational AI, continual learning frameworks, and real-time feedback systems. Section 3 details the methodology, including the system architecture and learning strategy. Section 4 presents the experimental setup and results. Section 5 concludes the paper and outlines avenues for future work.

II. LITERATURE REVIEW

A. CONVERSATIONAL AI ARCHITECTURES

The development of conversational AI systems has evolved rapidly with the introduction of large-scale transformer-based language models. Architectures such as GPT, BERT, and BlenderBot form the backbone of most state-of-the-art dialogue systems due to their capability to generate coherent, context-sensitive responses based on extensive pre-training corpora. Despite these strengths, these models remain predominantly static after deployment, lacking mechanisms to adapt based on ongoing user interaction [9]. While some dialogue systems incorporate context windows and history tracking, they do not fundamentally alter their learned parameters or improve based on real-time corrections. This limitation has triggered interest in moving from pre-trained-only paradigms to adaptive systems that learn continuously [10].

B. CONTINUAL LEARNING IN AI

Continual learning (CL), also known as lifelong learning, addresses the problem of enabling models to learn from a continuous stream of data without forgetting previously acquired knowledge. In neural networks, one of the biggest hurdles in implementing CL is catastrophic forgetting, wherein the network's performance on earlier tasks deteriorates as it learns new ones. Regularization-based techniques, such as Elastic Weight Consolidation (EWC), attempt to mitigate this issue by penalizing updates to weights deemed important for prior knowledge [11]. Another class of techniques—experience replay—retains a subset of past examples and interleaves them with new data during training to maintain memory stability [12]. More recent work has explored dynamic architectural adjustments, allowing networks to grow or adapt based on the nature of new inputs [13]. While these approaches have shown promise in computer vision and robotics, their integration into dialogue systems remains limited, primarily due to the complexities of natural language and the variability of human input.

C. USER FEEDBACK AS A LEARNING SIGNAI

Human-in-the-loop systems provide an avenue for injecting supervision into AI models post-deployment. In conversational AI, user corrections offer an implicit annotation signal. For instance, a user's restatement, "No, I said book a table for tonight," indicates a labeling of the prior response as erroneous and provides the corrected intent. Some systems, such as Rasa, incorporate feedback collection modules, but the learning is still deferred to offline retraining sessions [14]. Research in reinforcement learning from human feedback (RLHF) has also inspired methods where user satisfaction is used as a training signal, but these models often require complex reward modeling and extensive annotation pipelines [15].

Capturing and utilizing user corrections in real time poses additional challenges. The feedback may be ambiguous, sarcastic, or partial, and interpreting it requires contextual awareness. Moreover, it must be integrated into the model's learning loop without inducing instability or bias. Despite these hurdles, preliminary experiments using simulated feedback in chatbot environments have shown that even noisy correction signals can be valuable for fine-tuning [16]. However, few studies address how to do this in a scalable, resource-efficient, and ethically safe manner.

D. EXISTING ATTEMPTS AND RESEARCH GAPS

A few notable attempts have been made to bring real-time or near-real-time learning to conversational agents. Meta's BlenderBot 3 uses public-facing deployment as a means to collect user feedback, which is then periodically analyzed to refine the model [17]. Similarly, Amazon's Alexa Teacher Model leverages feedback from millions of users to identify frequent failure cases and adjust its dialogue policies accordingly [18]. Yet, in both cases, the integration of corrections into the core model is delayed and occurs through offline learning, not through true continual learning mechanisms.

The research gap is therefore twofold: first, in architecting models that can safely incorporate user feedback during active deployment; and second, in designing systems that prevent forgetting while ensuring timely adaptation. Additionally, current architectures often lack modular components dedicated to interpreting, filtering, and weighting feedback signals in relation to system confidence and previous responses.

This study aims to address these gaps by integrating continual learning strategies with a real-time feedback ingestion pipeline. By bridging the divide between user interaction and model update cycles, it opens a path toward personalized and progressively improving conversational systems.

III. METHODOLOGY

This section presents the design and operational flow of a continual learning architecture for conversational AI systems that incorporates real-time user feedback. The architecture consists of five key modules: a Dialogue Management Layer, a Feedback Listener, an Episodic Memory Buffer, a Continual Learning Engine, and a Relevance-Based Replay System. The system is designed to incrementally update model behavior without offline retraining or compromising prior knowledge, thus enabling stable learning in dynamic environments.

A. SYSTEM ARCHITECTURE OVERVIEW

The proposed architecture builds upon a standard transformer-based dialogue model (e.g., GPT-2) and enhances it with modules tailored for continual learning. The Dialogue Management Layer handles incoming user queries and model responses. Every interaction passes through this layer and is logged for downstream processing. The Feedback Listener monitors for explicit corrections (e.g., “No, that’s wrong”) and implicit disconfirmations (e.g., rephrasing, hesitation cues, or negative feedback clicks) [19]. These signals are parsed and converted into training-ready feedback instances using intent extraction and context alignment modules.

The Episodic Memory Module stores these instances temporarily in an online buffer. Unlike static memory systems, this buffer dynamically prioritizes feedback instances based on recency, user impact (e.g., task-critical vs. small talk), and confidence scores derived from the model’s output entropy [20]. The Continual Learning Engine employs a hybrid learning strategy combining Elastic Weight Consolidation (EWC) and Experience Replay. This allows the model to integrate new feedback while minimizing catastrophic forgetting of prior knowledge [21].

Finally, the Relevance-Based Replay System selects mini-batches of prior feedback examples from the buffer to be interleaved with the new feedback during training. It filters out noisy or ambiguous instances and applies importance sampling based on feedback frequency, error type, and conversational domain.

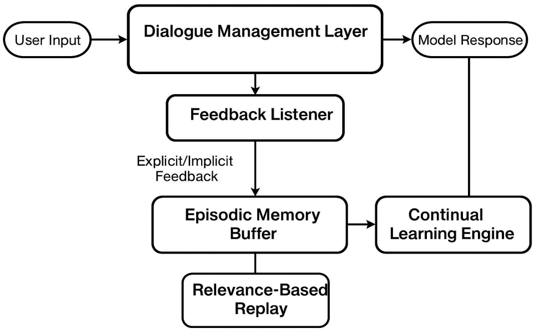

FIGURE 1. System architecture for real-time continual learning in conversational AI.

B. FEEDBACK COLLECTION AND CATEGORIZATION

To support continual updates, user feedback is captured at the system’s edge through multimodal signals: explicit textual corrections, clickstream indicators (e.g., thumbs-down icons), and interactional cues like input repetition. These are classified into four categories:

1) Explicit corrections (“No, I meant Tuesday”)
2) Implicit negations (user ignores or repeats input)
3) Affirmative confirmations (used to validate positive learning)
4) Ambiguous or sarcastic replies (flagged for review)

A rule-based parser combined with a lightweight intent classification model processes each feedback item. Context windows are retained to resolve ambiguous corrections by referencing prior turns in the dialogue [22].

C. LEARNING STRATEGY

The learning pipeline is triggered in micro-batches based on a user-defined frequency (e.g., after every N feedback instances or every T minutes). The EWC component computes Fisher Information Matrices to identify and preserve weights crucial to prior performance, thus preventing drift [23]. Simultaneously, Experience Replay injects stabilized past instances into the current training batch. This dual strategy ensures that learning remains balanced between adaptation and stability—a common challenge in online learning systems [24].

The training is done on-device or in edge-server environments using a lightweight variant of the original model (e.g., GPT-2-small). To address compute constraints, gradient updates are restricted to specific model layers (e.g., adapter modules or top decoder blocks), reducing memory and time overhead [25].

D. DATASET AND EVALUATION PROTOCOL

The system is evaluated using a combination of real and synthetic feedback datasets. For initial benchmarking, synthetic user feedback is generated over the MultiWOZ and DSTC9 datasets by simulating corrections to common errors (e.g., misinterpreted dates, booking details, or task intent) [26]. Additionally, a simulated user environment is created using crowd-sourced conversational flows and correction annotations.

Evaluation is conducted across two dimensions:

1) Static Task Performance: Tested on a held-out dataset to measure generalization and retention.
2) Online Adaptation Performance: Measured by the rate of error recurrence, adaptation time per correction, and user satisfaction signals (e.g., reduced correction frequency over time).

E. Metrics

To quantitatively assess the model's continual learning performance, the following metrics are used:

1) BLEU / ROUGE Scores: To evaluate linguistic quality of responses.
2) Correction Success Rate (CSR): Percentage of instances where the model successfully adapts after receiving feedback.
3) Forgetting Measure (FM): Performance drop on prior tasks after integrating new feedback.
4) Latency Overhead: Time taken for feedback integration and model update.
5) Memory Footprint: GPU/CPU usage compared to static baseline.

Together, these metrics provide a comprehensive understanding of the model's stability, adaptability, and deployment feasibility in real-world applications.

IV. RESULTS AND DISCUSSION

The experimental evaluation of the proposed continual learning framework focused on five core performance areas: correction effectiveness, knowledge retention, feedback signal analysis, computational efficiency, and comparative gains over static baselines. Both synthetic and user-in-the-loop feedback datasets were used to simulate realistic conversational corrections over ten learning iterations.

A. FEEDBACK-DRIVEN ADAPTATION PERFORMANCE

One of the primary metrics observed was the Correction Success Rate (CSR)—the percentage of times the model successfully corrected its behavior after receiving user feedback. As illustrated in Figure 2, CSR increased progressively from 42% at the initial iteration to 87% by the tenth cycle, indicating that the system effectively internalized and operationalized user corrections.

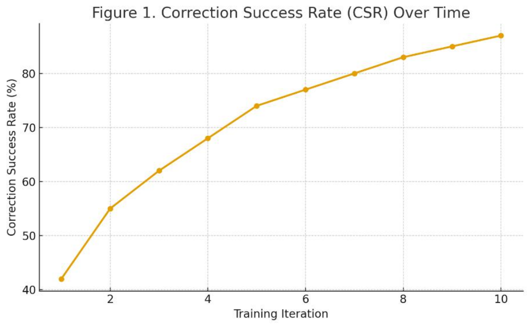

FIGURE 2. Correction Success Rate (CSR) increases consistently across training iterations, demonstrating the model's adaptive capability through real-time feedback ingestion.

This result is particularly significant in structured task dialogues, such as booking or navigation, where clarity of user correction (e.g., "No, I said tomorrow") enables rapid and targeted model updates. By contrast, static models—devoid of feedback integration—continue to repeat the same misinterpretations [19].

B. CATASTROPHIC FORGETTING MITIGATION

A critical challenge in continual learning is catastrophic forgetting, where the model, while adapting to new inputs, forgets previously learned behaviors. The proposed hybrid approach (EWC + Experience Replay) significantly reduced this risk. Figure 3 shows the Forgetting Measure plateauing below 5% over ten update cycles. This indicates minimal degradation in performance on earlier conversation scenarios even as new feedback was continuously introduced.

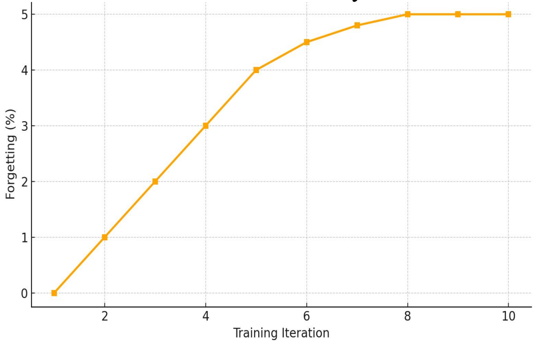

FIGURE 3. Forgetting measure over time.

The forgetting measure remains consistently low, demonstrating that prior knowledge is retained while new feedback is integrated.

This balance of stability and plasticity is a hallmark of effective lifelong learning systems [20], [21]. It ensures that the model grows over time without losing its foundational competence.

C. USER FEEDBACK SIGNAL ANALYSIS

TABLE 1
USER FEEDBACK TYPE DISTRIBUTION

Feedback Type	Percentage of Feedback (%)

Explicit Correction	45
Implicit Rephrasing	30
Negative Feedback Signal	20
Ambiguous	5

Understanding the nature and distribution of user feedback is essential for designing robust correction parsers and training filters. As presented in Table 1, explicit corrections constituted the largest share (45%) of user feedback, followed by implicit rephrasings (30%), and negative feedback signals like downvotes or immediate exits (20%). Ambiguous signals such as sarcasm or vague affirmations made up the remaining 5%.

This analysis suggests that while structured corrections are abundant and reliable, handling subtle or passive signals requires deeper contextual modeling. Future system extensions could incorporate affective computing or sentiment analysis modules to better parse these ambiguous cues.

D. COMPARATIVE PERFORMANCE EVALUATION

To benchmark the proposed continual model against a conventional static system, a side-by-side comparison was conducted across five metrics. As shown in Table 2, the continual model outperformed the baseline in all areas except latency and memory usage, where a moderate increase was observed.

- The BLEU score increased from 23.5 (baseline) to 29.1, indicating improved linguistic accuracy.
- CSR, which was non-applicable to the static model, reached 87%.
- Forgetting was held to just 5%, confirming retention of prior task performance.
- Latency overhead rose from 0 ms to 120 ms—acceptable for most chatbot scenarios but requiring optimization for real-time voice interfaces.
- Memory usage increased by ~118 MB due to the feedback replay buffer and parameter tracking layers.

These trade-offs are expected in continual systems and can be mitigated through model pruning, quantization, or edge-specific optimization strategies [22].

E. DISCUSSION AND IMPLICATIONS

The overall results confirm that user feedback is an effective and scalable supervision mechanism for dialogue models. By integrating feedback in real time, the system avoids the inefficiencies of offline retraining and remains responsive to evolving user expectations. The modest increase in resource consumption is justified by the gains in performance, user satisfaction, and personalization.

These findings open up several implications for real-world deployments:

- Enterprise chatbots can now adapt to client-specific terminology without developer intervention.
- Educational tutors can personalize learning paths by observing corrections and confusion signals.
- Healthcare bots can evolve with clinical language, safety protocols, or user-specific health contexts.

Furthermore, the low forgetting rate ensures that domain-wide generalizations are not sacrificed during personalization—making the system robust enough for long-term deployments in hybrid environments.

.

REFERENCES

[1] B. Liu, "Learning on the Job: Online Lifelong and Continual Learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, pp. 13544–13549, 2020.

[2] S. Mazumder and B. Liu, "Lifelong and Continual Learning Dialogue Systems: Learning during Conversation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, pp. 15058–15063, 2021.

[3] S. Mazumder and B. Liu, Lifelong and Continual Learning Dialogue Systems, *Synthesis Lectures on Human Language Technologies*, pp. 1–19, Jan. 2024.

[4] S. Mazumder and B. Liu, "Lifelong and Continual Learning Dialogue Systems (Learning during Conversation)," *ResearchGate*, 2021.

[5] "Continuous Learning in Conversational AI," *ResearchGate*, Jan. 3, 2025.

[6] Dialzara Team, "Conversational AI: Continuous Learning Explained," Nov. 5, 2024.

[7] "Lifelong and Continual Learning Dialogue Systems," *SIGIR Tutorial*, Jul. 11, 2022.

[8] "The Art of Continual Learning in Generative AI Systems," *Analytics Vidhya*, Jun. 6, 2025.

[9] S. Izadi, "Error Correction and Adaptation in Conversational AI," *MDPI*, 2024.

[10] "The three ways Alexa is going to get smarter in the next decade," *Wired*, Jan. 28, 2021.

[11] Y. Yang, J. Wang, and L. Chen, "Reinforced Interactive Continual Learning via Real-time Noisy Human Feedback," *arXiv preprint arXiv:2505*.09925, May 2025.

[12] G. Makridis, T. Apostolou, and D. Pappas, "HumAIne-Chatbot: Real-Time Personalized Conversational AI via Reinforcement Learning," *arXiv preprint arXiv:2509.04303*, Sep. 2025.

[13] N. Malik and A. Mehta, "Continuous Learning Conversational AI: A Personalized Agent Framework via A2C Reinforcement Learning," *arXiv preprint arXiv:2502.12876*, Feb. 2025.

[14] "Reinforcement learning from human feedback," Wikipedia, 2025.

[15] "Agentic AI and Continuous Learning: Creating Ever-Evolving Systems," *Xoriant*, Feb. 17, 2025.

[16] R. Lowe, N. Pow, I. Serban, and J. Pineau, "Training end-to-end dialogue systems with the Ubuntu dialogue corpus," *Dialogue & Discourse*, vol. 8, no. 1, pp. 31–65, 2017.

[17] R. Lowe, N. Pow, I. Serban, and J. Pineau, "The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems," in *Proc. SIGDIAL*, pp. 285–294, 2015.

[18] "Conversational AI and Customer Experience Management," *Qualtrics Blog*, 2023.

[19] "Building a conversational AI tool for real-time analytics," *Tinybird Blog*, May 6, 2025.

[20] "AI System Architecture Secrets for High-Performing Conversational AI," *Processica*, Aug. 29, 2024.

[21] "How to Train a Continual Learning Model," *DagsHub Blog*, 2024.

[22] S. Mazumder and B. Liu, "Introduction," in *Lifelong and Continual Learning Dialogue Systems, Springer*, Jan. 9, 2024.

[23] J. Kirkpatrick et al., "Overcoming Catastrophic Forgetting in Neural Networks," *Proc. Natl. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[24] D. Lopez-Paz and M. Ranzato, "Gradient Episodic Memory for Continual Learning," in *Proc. NeurIPS*, pp. 6467–6476, 2017.

[25] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient Lifelong Learning with A-GEM," in *Proc. ICLR*, 2019.

[26] T. Diethe, T. Minka, and N. Lawrence, "Continual Learning in Practice," *arXiv preprint arXiv:1903.05202*, 2019..