Adaptive Self-Learning Agentic AI System: A Continuous Fine-Tuning Framework for Speech-to-Text Models

Kavya Venkatesh
Computer Science
Columbia University
New York, USA
kv2458@columbia.edu

Gautam Agarwal
Computer Science
Columbia University
New York, USA
ga2726@columbia.edu

Shivangi Kumar
Computer Science
Columbia University
New York, USA
sk5659@columbia.edu

***Abstract*—This project proposes an adaptive self-learning Agentic AI framework designed to continuously fine-tune generative models through autonomous error detection, correction, and re-training. The system aims to build a closed feedback loop that integrates a correction agent, fine-tuning pipeline, and adaptive scheduling algorithm. Demonstrated through a Speech-to-Text (STT) use case, the project emphasizes model self-improvement, performance monitoring, and generalization to broader generative AI tasks. The implementation includes automated hyperparameter optimization, fine-tuning orchestration, and adaptive scheduling to optimize learning frequency while mitigating overfitting and computational cost.**

***Index Terms*—Agentic AI, Fine-Tuning, Adaptive Scheduling, Speech-to-Text, Generative AI, Continuous Learning**

I. Introduction

Modern generative AI models often operate in static deployment paradigms where they fail to learn from their own mistakes after release. This leads to performance degradation over time and necessitates manual fine-tuning. To address this limitation, we propose a self-learning Agentic AI system capable of autonomous improvement through continuous feedback and dynamic retraining.

The proposed framework demonstrates these principles using a Speech-to-Text (STT) model. The system integrates an LLM-based correction agent, automated fine-tuning pipelines, and adaptive scheduling mechanisms that decide when to retrain based on performance metrics. Over time, the model improves continuously with minimal human intervention, balancing computational efficiency and performance gains.

II. Project Overview

A. Motivation and Objectives

The motivation stems from the inefficiency of static AI systems and the high cost of manual retraining. This project aims to:

- Develop a closed-loop self-learning system for AI model improvement.
- Integrate LLM-based agents for autonomous output correction.
- Automate fine-tuning and hyperparameter preparation.
- Implement adaptive fine-tuning frequency mechanisms to prevent overfitting.
- Generalize the framework to other generative AI tasks beyond STT.

B. Research Questions

1) How can an AI system autonomously identify and correct its own errors?
2) What strategies enable optimal fine-tuning frequency balancing accuracy and computational cost?
3) Can this framework generalize effectively across domains beyond STT?

III. Related Work

Recent advances in AI have positioned agentic systems as the next frontier in autonomous decision-making, enabling self-correction through iterative generate-critique-improve cycles. Modern speech recognition models like Whisper (680k hours training, 8-20% WER) and Wav2Vec2 (4.8% WER on clean speech) achieve strong baseline performance but benefit substantially from domain-specific fine-tuning to address error patterns in specialized contexts.

Traditional fine-tuning approaches face two critical challenges: computational expense and catastrophic forgetting (where models lose previously learned knowledge during adaptation). Parameter-efficient methods such as LoRA address the first challenge by reducing trainable parameters by 10,000× while maintaining performance comparable to full fine-tuning. To combat catastrophic forgetting, state-of-the-art approaches employ replay-based methods and parameter regularization to maintain retention while learning new patterns. Recent adaptive optimization techniques further enhance efficiency by dynamically adjusting hyperparameters based on performance feedback, enabling smaller models to outperform larger ones.

Complementing model-level improvements, LLM-based error correction systems have emerged as powerful post-processing solutions. By using N-best hypothesis rescoring and rule-constrained prompting, these systems achieve 10-20% relative WER reductions across diverse domains without task-specific training. However, existing approaches treat error correction and model fine-tuning as separate processes.

Our framework uniquely integrates these components into a unified closed-loop system where corrections directly inform adaptive fine-tuning decisions. With performance-aware scheduling and cost-efficiency optimization, we present the first fully autonomous self-learning architecture for continuous STT improvement with generalization to other generative tasks.

IV. Phased Implementation Plan

A. Phase 1: Baseline Model and Evaluation

Select and deploy a cost-effective pre-trained STT model such as Whisper or Wav2Vec2. Establish benchmarks using Word Error Rate (WER), Character Error Rate (CER), and latency metrics. Develop a dataset focusing on difficult transcription cases (e.g., background noise, accents).

B. Phase 2: LLM-Based Correction Agent

Integrate an LLM agent capable of detecting and correcting transcription errors. The agent compares baseline outputs with expected linguistic patterns and flags cases requiring improvement. Corrected outputs are logged into a fine-tuning dataset with metadata tracking performance improvements.

C. Phase 3: Automated Fine-Tuning Pipeline

Implement a library for automated hyperparameter optimization (learning rate, batch size, epochs). Fine-tune the model after accumulating n error cases. The new model version is validated against the baseline using standardized evaluation sets. If it surpasses baseline metrics, it replaces the existing model.

D. Phase 4: Adaptive Scheduling

Develop an adaptive scheduling algorithm that adjusts the fine-tuning threshold n dynamically based on recent model improvements. When accuracy gains diminish, n increases to reduce retraining frequency, preventing overfitting and conserving resources.

E. Phase 5: Generalization and Testing

Abstract the system to support other generative tasks such as text generation or translation. Demonstrate the framework's modularity and scalability through a secondary application domain.

V. System Architecture

The proposed system architecture (Fig. 1) consists of multiple layers:

1) **Inference Layer:** Handles real-time user input and transcription output.
2) **Correction Layer:** Employs an LLM to assess and improve outputs.
3) **Data Management:** Stores failed cases and prepares fine-tuning datasets.
4) **Fine-Tuning Orchestration:** Automates training and validation cycles.
5) **Adaptive Scheduling Layer:** Adjusts retraining frequency and manages model versions.

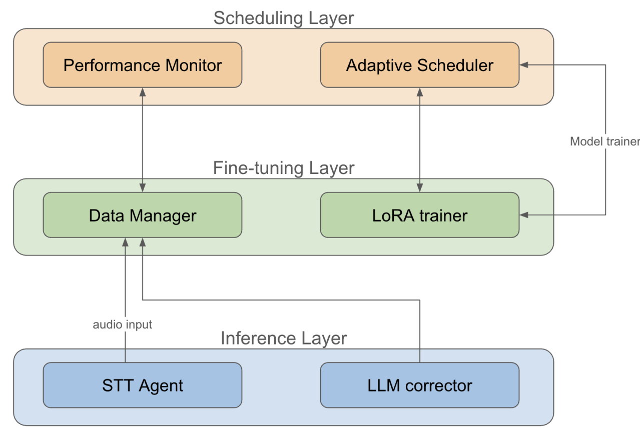

Fig. 1. Overview of the self-learning Agentic AI architecture.

VI. Data Points and Metrics

A. Model Performance Metrics

- Word Error Rate (WER) and Character Error Rate (CER)
- Latency and throughput during inference
- Computational cost (GPU-hours, API calls)

B. Agent Performance Metrics

- Correction accuracy and consistency
- False positive rate in corrections
- Confidence scores and runtime latency

C. Fine-Tuning Data Metrics

- Dataset size and diversity
- Hyperparameter configurations
- Improvement per fine-tuning iteration

VII. Evaluation Strategy

The project will be evaluated across four main dimensions:

1) **Quantitative Analysis:** Compare pre and post-fine-tuning WER/CER improvements and statistical significance (paired t-tests).
2) **System Efficiency:** Measure cost per accuracy gain, convergence time, and computational efficiency.
3) **Ablation Studies:** Evaluate impact of correction agent, adaptive scheduling, and hyperparameter tuning independently.
4) **Qualitative Assessment:** Review corrected transcripts for semantic accuracy and linguistic fluency.

VIII. Expected Outcomes

- A functional, self-improving STT system demonstrating measurable accuracy gains.
- Automated hyperparameter tuning and retraining pipeline.
- Adaptive fine-tuning frequency scheduler to balance performance and cost.
- Generalized framework applicable to multiple GenAI domains.

IX. Risk and Mitigation

- **Overfitting:** Mitigated using adaptive n scheduling and validation sets.
- **Resource Costs:** Controlled via lightweight fine-tuning methods like LoRA.
- **Model Degradation:** Avoided through regression testing before deployment.

X. Conclusion

This project builds a self-learning Agentic AI system that autonomously improves through adaptive fine-tuning. Demonstrated on Speech-to-Text, it introduces a novel blend of agentic correction, automated optimization, and adaptive scheduling. The modular design ensures scalability, reproducibility, and generalizability across multiple generative AI applications.

Acknowledgment

The authors thank the Columbia University Generative AI course staff for guidance and feedback on the project framework.

References

[1] M. Young, *The Technical Writer's Handbook*, University Science, 1989.
[2] OpenAI, "GPT-4 Technical Report," 2023.
[3] A. Radford et al., "Whisper: Robust Speech Recognition via Large-Scale Weak Supervision," OpenAI, 2022.
[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, 1997.
[5] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," EMNLP 2020.
[6] A. Ng, "Agentic AI: Self-Reflection and Autonomous Systems," AI Index Report, 2024.
[7] J. Madaan et al., "Self-Refine: Iterative Refinement with Self-Feedback," NeurIPS, 2023.
[8] E. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," ICLR, 2022.
[9] Y. Fan et al., "Adaptive Divergence Regularized Policy Optimization," NeurIPS, 2025.
[10] A. Baevski et al., "wav2vec 2.0: Self-Supervised Learning of Speech," NeurIPS, 2020.