

Llm-Based Generative Error Correction For Hindi And Punjabi Asr Using In-Context Learning And Task-Activating Prompting

Monika¹, Vishal Passricha²

¹Department of Computer Science and Engineering, Central University of Haryana,
Mahendergarh, India

²Department of Computer Science and Engineering, Central University of Haryana, Mahendergarh,
India

Abstract: With the rapid development of Automatic Speech Recognition (ASR) technology, end-to-end (E2E) speech recognition systems have gained significant attention due to their ability to directly convert raw speech signals into text. However, such systems heavily rely on large amounts of labeled speech data, which severely limits model training performance and generalization. To address these limitations, Large Language Models (LLMs) have emerged as a powerful substrate for Generative Error Correction (GER), which provide the semantic and phonetic reasoning necessary to refine acoustic hypotheses. This paper proposes a novel E2E ASR framework that utilizes a deep-fusion strategy to tightly couple an LLM with an E2E decoder through contextualized prompting. Unlike traditional methods that require resource-intensive fine-tuning, our approach applies In-Context Learning (ICL) and Task-Activating Prompting (TAP) to guide the LLM through a step-by-step reasoning process. This allows the model to synthesize corrections for tokens entirely absent from the initial N-best list. It effectively surpassing the oracle WER upper bound that limits conventional re-ranking strategies. Experimental results demonstrate that proposed framework significantly improves recognition accuracy and stabilizes outputs under low-resource conditions. By shifting the paradigm from discriminatory rescoring to generative synthesis, we provide a reproducible evaluation benchmark and release as pre-trained models to facilitate future research in LLM-enhanced ASR.

Keywords: Generative Error Correction, In-Context Learning, Large Language Models, Low-resource Speech Processing, Task-Activating Prompting.

I. INTRODUCTION

The paradigm of automatic speech recognition (ASR) has shifted from hybrid architectures toward end-to-end (E2E) models [1, 2]. Unlike traditional systems that maintain separate acoustic and language modeling components, E2E approaches rely on a single feedforward neural network to map speech signals directly to text sequences. E2E approaches simplify the recognition pipeline by merging acoustic and language model and achieve gains in overall performance and robustness. Although this architecture introduces distinct architectural challenges regarding alignment and dependency modeling.

Current E2E research mainly focuses on three key approaches: Connectionist Temporal Classification (CTC) [3, 4], Recurrent Neural Network Transducers (RNN-T) [5, 6], and Attention-based Encoder-Decoder (AED) models [7, 8]. While all three paradigms treat ASR as a sequence-to-sequence problem, their mechanisms differ significantly. CTC is preferred for its computational efficiency and independent token prediction, but its conditional independence assumption limits accuracy for complex utterances. To address this issues, RNN-T introduces an autoregressive decoder to condition predictions on previous outputs, however, this often comes at the cost of training instability and high memory consumption [9]. Similarly, AED models jointly apply encoders and decoders to leverage full context, but they introduce the latency and remains sensitive to noise-induced alignment degradation [10].

Despite the capability of E2E systems to estimate the posterior probability $P(W|X)$ directly, where X is the speech signal and W is the corresponding text. External Language

Models (LMs) remain essential for guiding recognition toward linguistically plausible results. Earlier, ASR was framed as a noisy-channel problem seeking to maximize:

$$\hat{W} = \arg \max_W P(X|W) P(W) \quad (1)$$

In this Bayesian framework, the language model $P(W)$ is trained independently on text data to complement the acoustic model [11]. This structure is still persisted into the deep learning era. Even in modern E2E pipelines, LMs are frequently integrated via fusion strategies like shallow or cold fusion [12, 13] or applied in second-pass rescoring to re-rank beam search hypotheses.

The advent of Large Language Models (LLMs) has expanded this role from simple probability estimation to semantic reasoning, bidirectional rescoring, knowledge distillation, and post recognition error correction [14-16]. An encouraging area in this domain is Generative Error Correction (GER), where LLMs with Low-rank adaption (LoRA) adapter map ASR hypotheses directly to final transcriptions [17-20]. However, using large language models usually requires heavy fine-tuning so that these models can work effectively for the ASR task. This work takes advantage of this capability and considers integrating audio modality into LLMs.

To address the computational overhead of fine-tuning, we explore In-Context Learning (ICL) and Task-Activating Prompting (TAP). ICL allows pre-trained models to recognize patterns implicitly encoded in prompts without weight updates, while TAP guides the model through step-by-step reasoning rather than direct instruction [18, 21]. In this work, we propose a two-pass E2E ASR framework utilizing an RNN-T encoder-decoder and a full-context LLM-based GER. We replace traditional fine-tuning process with prompt-engineering strategies to enhance ASR rescoring. The primary contributions of this work are as follows:

- **LLM-based GER:** The proposed model architecture evaluates the utility of LLMs to optimize the decoding process within an E2E system to enhance recognition rate.
- **Zero-Shot Correction:** Two pass-ASR systems based on LLM is proposed to predict true transcriptions from N-best hypotheses. Crucially, we replace the fine-tuning process with In-Context Learning and Task-Activating Prompt strategies.
- **Generative Error Paradigm:** In proposed architecture introduced for generative error correction, the ASR generates word-piece elements and the LLM utilizes linguistic and contextual knowledge to correct the final sequence [22].

II. Related Work

Over the years, researchers have proposed a wide range of approaches, including traditional ASR systems, E2E systems, ML-based system to effectively recognize the human speech. With the rise of deep learning models, such as CNNs, RNNs etc. learn high level feature from the raw speech signals and regime is changing.

2.1 ASR Rescoring and Error Correction

The field of ASR rescoring and error correction has evolved rapidly in speech processing. It started with discriminatory selection and presently use generative synthesis [23]. Earlier, the field relied on n-gram language model rescoring, which functions by re-ranking an N-best hypothesis list to select the most probable solution. As neural network based LM introduced, researchers started replacing traditional statistical models with pre-trained Language Models (LMs) [24, 25]. In these approaches, unidirectional models like GPT-2 compute log-likelihoods, while bidirectional models such as BERT [26] and RoBERTa [27] utilize pseudo-log-likelihoods to score hypotheses.

However, the emergence of Large Language Models (LLMs) has given the new heights. Rather than merely selecting the best option among N-best hypotheses, modern

approaches utilize LLMs to directly generate corrected transcriptions using the full N-best list as context [28]. Transformer-based models facilitate this by processing N-best hypotheses in parallel, allowing them to simultaneously rescore and rewrite text [29].

This generative capability of LLMs are giving new heights to ASR research. It overcomes the error. Traditional methods like industry-standard Recognizer Output Voting Error Reduction (ROVER) [30, 31] are limited by the quality of the N-best list; if the correct token is missing, they cannot recover it. On the other hand, LLMs apply phonetic inference and contextual reasoning to correct errors. They insert tokens absent from the original hypotheses [32]. Recent benchmark, HyPoradise demonstrates the power of this paradigm by achieving up to 15% Word Error Rate (WER) reduction in targeted domains [17]. The convergence of reproducible evaluation, multimodal integration, and parameter-efficient tuning has strongly support LLMs for generative error correction that surpasses traditional re-ranking bounds.

2.2 Prompt Engineering

In recent years, various methodology has been proposed for adapting LLMs to new tasks [33-35]. Fine-tuning is computationally expensive process which updates the parameters. In-Context Learning (ICL) enables pre-trained LLMs to perform novel tasks at inference time by learning from examples embedded directly within the input prompt [36-39]. This mechanism operates on the principle of analogy. Prompt is used as a semantic prior to guide the model's reasoning without requiring any spurious corrections [28, 37]. When a task description and supporting demonstrations are formulated as a natural language prompt, they essentially prime the model's chain of thought [36, 38].

However, the use of ICL in ASR systems is not without challenges. In Naive direct prompting, a model is simply asked to fix this sentence. The generated results are often leads to instability, hallucinated confidence, or unnecessary stylistic edits [39, 40]. To mitigate this shortcoming, Task-Activating Prompting (TAP) has emerged as a robust variant for error correction [18]. By structuring prompts to guide the model through progressively complex examples, TAP encourages disciplined reasoning and reduces spurious corrections [41]. These findings highlights that while pre-trained models possess the requisite knowledge for error correction. Performance strongly depends on designing the prompts carefully.

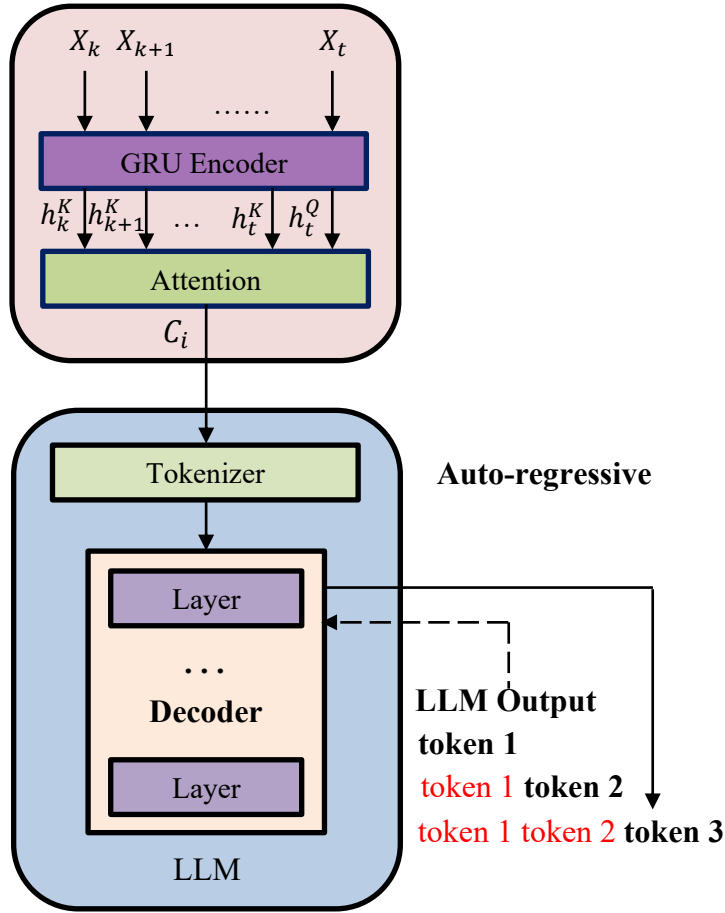
III. LLMs-based Generative Error Correction ASR Model via ICL and TAP prompting

Generative Error Correction (GER) serves as the refinement stage in ASR pipeline which address common ASR failures such as accent variation, environmental noise, and lexical ambiguity. Traditional rescoring methods merely select the most correct option from an N-best list. On the other hand, GER applies text-to-text modeling to rewrite and synthesize the correct transcription [18, 19]. By resolving semantic inconsistencies and repairing grammatical structures, the final output quality does not depend on the acoustic model's initial hypothesis.

3.1 System Architecture

The proposed architecture is illustrated in Figure 1. The proposed architecture functions as a dual-module system which comprising a RNN-T Acoustic Encoder Decoder and an LLM-driven GER. A core challenge in this design is aligning the acoustic feature space with the disparate token vocabularies of source LLMs.

Figure 1: Proposed ASR correction architecture with LLM-driven generative error correction.



3.2 Acoustic Encoder

The encoder's task is to extract high-level informative features from the raw speech spectrum. The input features $\mathbf{X}$ are first projected through a non-linear transformation to produce an intermediate representation suitable for recurrent processing. Formally, the transformed input $\tilde{\mathbf{x}}(t)$ is obtained as:

$$\tilde{\mathbf{x}}(t) = \sigma(W_{in}\mathbf{x}_t + b_{in}) \quad (2)$$

This representation is fed into GRUs to model temporal dependencies. GRUs are preferred over Long Short-Term Memory (LSTM) networks due to their superior computational efficiency and comparable performance in sequential modeling tasks [42]. The hidden state h_t is updated recursively:

$$h_t = \text{GRU}(\mathbf{x}_t, h_{t-1}) \quad (3)$$

Here, $\text{GRU}(\cdot)$ encapsulates the gating mechanisms that regulate information flow. The encoder output is then split into key (K) and value (V) representations to drive the attention mechanism:

$$K = W_k h, \quad V = W_v h$$

3.3 Attention Mechanism

An attention mechanism is applied to bridge the modality gap between continuous speech frames and discrete text tokens. This allows the model to form fixed-length context

vectors by weighing the relevance of specific encoder states against the current decoder state. The context vector c_t at decoding step t is computed as a weighted sum of the encoder states:

$$c_t = \sum_{j=1}^L \alpha_{tj} h_j \quad (4)$$

where α_{tj} represents the attention weights. To balance accuracy with latency, we implement two distinct attention strategies: i. In Causal Dynamic Attention, weights are computed using the entire history of the input sequence. While this captures rich global context, the contribution of distant frames often diminishes in long utterances. ii. Causal Local Attention strategy restricts the attention window to a localized neighborhood of frames which reduces computational cost. It focuses the model on immediate acoustic triggers.

3.3.1 Unidirectional attentional-GRU encoder

A gated recurrent unit (GRU) is a recurrent neural network architecture having gating mechanisms to control the flow of information across time steps. Unidirectional GRU processes the information in a single direction and is widely used in sequence-to-sequence learning. As compared to LSTM networks, GRUs contain fewer parameters which mitigate the vanishing gradient issues. This study uses a modified GRU in which the conventional hyperbolic tangent ($\tanh$) activation is replaced by attention-based ReLU (aReLU) [43]. The $\tanh$ function involves dense nonlinear computations that increase activation complexity. Whereas, aReLU introduces learnable data-adaptive parameters that selectively amplify positive activations while suppressing negative responses. The attention mechanism embedded in aReLU learns element-wise residuals within the active region which enables more stable gradient propagation during training. The training process remains robust towards vanishing gradient since attention mechanism in aReLU activation learns element-wise residuals of the active region. The attention activation through aReLU leads to well-focused activation in significant areas of the feature map. Having additional learnable parameters (α and β) per layer enables faster training at lower learning rates. aReLU is denoted as $f(x_i, \alpha, \beta)$ using a combination of an element-by-element sign-based attention approach $l(x_i, \alpha, \beta)$ and classical ReLU $R(x_i)$, as follows:

$$f(x_i, \alpha, \beta) = l(x_i, \alpha, \beta) + R(x_i) \quad (5)$$

$$f(x_i, \alpha, \beta) = \begin{cases} D(\alpha)x_i & \text{when } x_i < 0 \\ (1 + \sigma(\beta))x_i & \text{when } x_i > 0 \end{cases} \quad (6)$$

where $X = x_i$ is input to the activation layer, $[\alpha, \beta] \in \mathbb{R}^2$, $D(\cdot)$ grasps the input variables to $[0.01, 0.99]$ such that preventing α to be zero, and σ shows the sigmoid activation. The inclusion of aReLU in GRUs can assist in capturing long-term contextual dependencies between features. As a result, in addition to preventing gradient vanishing, the use of aReLU in the GRU can aid in capturing these long-term dependencies and improve the performance of ASR system.

3.4 Decoding and Hypothesis Refinement

The decoding phase reconstructs the speech content by integrating the attention-derived context vectors with the decoder's internal state. Beam Search is employed to explore the hypothesis space efficiently. Given the context vector c_t and the previous decoder state s_{t-1} , the decoder generates the output probability distribution for the next token y_t :

$$P(y_t | y_{<t}, X) = \text{Softmax}(W_{\text{out}}[c_t; s_t] + b_{\text{out}}) \quad (7)$$

This process maximizes the likelihood of the target sequence W given the noisy input X . In our framework, these initial hypotheses serve as the prompt-basis for the subsequent LLM-based generative correction.

3.4.1 Hypothesis Refinement

AED –based ASR + LLM-based correction is architecturally closed with Google’s two pass ASR [44]. LLM as rescoring model choose the best hypothesis among the given N-best hypotheses. It offers strong gains without hallucination. The use LLM as post-decoder is safe and effective. It performs GER and Text-to-text refinement.

$$\hat{W} = \arg \max_{W_i \in N\text{-best}} (\log P_{\text{ASR}}(W_i|X) + \lambda \log P_{\text{LLM}}(W_i)) \quad (8)$$

The LLM remains frozen, but its behavior adapts through iterative, Task-activating prompting conditioned on ASR hypotheses, confidence signals, and historical corrections. In direct prompting, the ASR system generates an N-best list of candidate transcriptions, and the language model selects the most probable one. However, this approach often results in over-correction or confidence hallucination. To overcome this limitations, we introduce an in-context learning framework that integrates large language model into the ASR decoding pipeline without retraining. Instead of applying static post-hoc correction, the LLM dynamically adapts its behavior through structured prompts constructed from N-best ASR hypotheses and confidence signals. This enables context-aware reasoning over competing transcriptions while preserving acoustic faithfulness. In-context learning turns the LLM from a passive text corrector into an active reasoning partner which is guided by ASR uncertainty and hypotheses, while leaving acoustic modeling untouched.

Construct an interactive prompt: Instead of feeding only the 1-best hypothesis, we provide structured context:

You are assisting as a Hindi speech recognition system. Below are multiple transcription hypotheses generated from speech. Some may contain recognition errors due to noise or pronunciation.

Hypotheses:

- 1) भारत एक देश है
- 2) भारत एक देश हे
- 3) भारत एक देस है

Task: Compare the hypotheses, synthesize the most acoustically and linguistically plausible transcription. Do not introduce new content and preserve the speaker’s original intent

Task-Activating Prompting: The ASR system is uncertain. Carefully reason over all hypotheses before deciding.

Final answer: भारत एक देश है

First LLM pass selects best hypothesis. Second pass verifies consistency and third pass performs final cleanup. This is similar to human proofreading.

Algorithm 1: Interactive In-Context Learning-based Speech Recognition Refinement**Input:**

- Raw Speech Signal X
- Trained end-to-end ASR model $\mathcal{M}_{ASR}$
- Beam width B
- Pre-trained Large Language Model $\mathcal{M}_{LLM}$

Output:

- Final refined transcription $\hat{W}$

Steps**1. Acoustic Feature Extraction**

Extract log-Mel spectrogram features F from the input speech signal X .

2. End-to-End Acoustic Encoding

Pass F through the ASR encoder to obtain frame-level acoustic representations:

$$\mathbf{H} = \text{Encoder}(\mathbf{F})$$

3. Initial Hypothesis Generation

Apply beam search decoding with beam width B to generate an N-best hypothesis set:

$$\mathcal{H} = \{(W_1, s_1), (W_2, s_2), \dots, (W_B, s_B)\}$$

where W_i and s_i denote the i^{th} hypothesis and its corresponding score.

4. Interactive Prompt Construction

Construct an in-context learning prompt by organizing the N-best hypotheses as structured contextual examples:

$$P = \text{Prompt}(\mathcal{H})$$

The prompt instructs the LLM to resolve errors related to grammar, word choice, and contextual consistency without altering semantic intent.

5. LLM-based Hypotheses Refinement

Feed the constructed prompt P into the frozen Large Language Model:

$$\hat{W} = \mathcal{M}_{LLM}(P)$$

6. Final Transcription Selection

Output the refined transcription $\hat{W}$ as the final ASR result.

Return $\hat{W}$

IV. Experiments and Results

This section outlines the dataset used, language model configuration, hardware specifications, and experimental configuration used in this study. We evaluate the performance of proposed GER correction model for distinct LLMs on two different datasets.

4.1 Dataset Analysis

To validate the robustness and generalizability of the proposed model, the experimental evaluation of the proposed model is conducted on two different datasets. The first evaluation is performed using Hindi dataset to verify the core architecture and its performance while the second evaluation focuses on Punjabi speech corpus to assess the model's performance on tonal and dialectal variations.

4.1.1 Hindi Corpus

The Hindi dataset serves as the primary verification set, consisting of more than 10 million words. The training phase uses 8,253,742 words from 392 different speakers, while 2,063,435 words from 96 different speakers were taken for testing. This large-size dataset ensures the model can handle the phonetic complexity of a major Indo-Aryan language. Table 1 shows the composition structure for Hindi speech corpus.

Table 1: Hindi Speech corpus composition structure

Subset	Participants	Volume (words)
Training Set	392	8,253,742
Testing Set	96	2,063,435
Total	488	10,317,177

4.1.2 Punjabi Tonal Corpus

The diversity of the training data is crucial for the model's ability to generalize across different phonetic and prosodic environments. A key contribution of this research is the use of a Punjabi speech corpus, which is particularly well-recognized for its ability to reflect the unique tonal characteristics of the language [45]. This 65-hour corpus is recorded from 180 native speakers across the Malwa, Majha, Doaba, and Powadh regions which captures the rich dialectal diversity of Punjab. Sentences are specifically designed to be phonetically rich to ensure that model encounters a wide array of consonant-vowel combinations. A mix of affirmative, interrogative, and doubtful sentences is considered. 100 dedicated tonal words and 5,000 high-frequency terms are covered to test the system's sensitivity to pitch-based meaning. Table 2 shows the composition structure for Punjabi speech corpus.

Table 2: Punjabi Speech corpus composition structure

Subset	Participants	Volume (utterances)
Training Set	144	95,600
Testing Set	36	23,900
Total	180	119,500

4.2 Language Model Configuration

This study uses two distinct LLM frameworks to evaluate performance: the generative GPT-4o mini and the retrieval-augmented Perplexity AI.

4.2.1 GPT-4o-mini

GPT-4o mini represents a large-scale, multimodal generative framework specifically optimized for reasoning-intensive tasks and complex instruction following. Built upon an advanced transformer architecture, the model features enhanced mechanisms for long-range dependency modeling and cross-modal reasoning supporting both text and vision. Its training regimen integrates supervised fine-tuning, and Reinforcement Learning from Human Feedback (RLHF) to ensure high robustness and factual consistency.

With a parameter count approximately 2 trillion, GPT-4o mini employs adaptive decoding strategies that allow for dynamic trade-offs between deterministic accuracy and creative generation. Tightly integrated safety layers mitigate bias and hallucination without compromising analytical depth. For this experiment, we use the GPT-4o-mini variant with task-activated prompting to conduct zero-shot and few-shot evaluations via designed task prompts [46].

4.2.2 Perplexity AI

Unlike monolithic language models, Perplexity AI operates on a Retrieval-Augmented Generation (RAG) architecture. This system synthesizes the generative capabilities of large, pre-trained transformers with a real-time web retrieval layer to produce grounded responses. The core architecture dynamically routes user queries to optimal foundation

models based on task complexity and latency constraints. Relevant documents are retrieved from curated web sources, embedded, ranked, and injected directly into the model's context window to guide generation. This tight coupling of retrieval and synthesis prioritizes factual density and explicitly minimizes hallucination. By enforcing source attribution and adaptive summarization, Perplexity is particularly configured here for research-oriented information access and exploratory analysis.

4.3 Computational Infrastructure

The designed ASR architecture is evaluated on a supercomputer named PARAM Shavak Yuva-II, which features two multicore CPUs with 18 cores each, accompanied by two GPGPU accelerator cards. The system is equipped with 64GB of RAM, 8TB of storage, and incorporates Nvidia Pascal architecture-based co-processing technologies. Running on Ubuntu v20.04, the computer is configured with a deep learning GPU software environment, and the Anaconda toolkit is employed for implementation using PyTorch.

4.4 Experimental Configuration

For both languages, the models were trained separately using a consistent set of hyperparameters to ensure a fair comparative analysis. We employed Adam optimizer to update the network parameters. It maintains a learning rate of 1×10^{-4} . All training was conducted at an 8-bit precision level to balance computational efficiency with model accuracy. We employ a batch size of 8 with the regularization hyperparameter λ set to 10^{-4} . The model undergoes a total of 15 training epochs to ensure convergence without overfitting. To maintain consistency across linguistic contexts, the same English prompts provided in the repository is used for both training and testing phases without modification. Table 3 represents the details of the hyperparameter used with their value and experimental settings used for Hindi and Punjabi datasets.

Table 3: Comparative Experimental Settings for Hindi and Punjabi Datasets

Dataset	Hindi	Punjabi
Total Duration	121 Hrs.	65 Hrs
Speakers	488 (234 Female + 254 Male)	180 (74 Female + 106 Male)
Train/Test Split	80% / 20%	80% / 20%
Bit-Precision	8-bit	8-bit
Epochs	15	15
Learning rate	1e-4	1e-4
Low-Rank Adaptation	4	4
Prompt Language	English	English

4.5 Experimental Results and Analysis

For ASR models, the most important performance metric is word error rate (WER). In practice, WER determine user experience. WER means the primary accuracy metric. Lower percentages indicate better transcription accuracy. A 4% WER means the model makes 1 error per 25 words on average. The proposed architecture performance is evaluated in WER. To evaluate the generative error correction capabilities, we configured the systems as follows:

4.5.1 Baseline Hindi and Punjabi Experiments

For the Hindi dataset GER experiment, pre-trained versions of the GPT and Perplexity models are utilized. These models were deployed in their base state to establish a performance benchmark, leveraging their inherent multilingual capabilities without task-specific architectural modification. For the Punjabi dataset GER experiments, also pre-trained GPT and Perplexity models are used. To preserve the integrity of the tonal features, speed perturbation was explicitly excluded from the augmentation pipeline. By validating the system on both the Hindi speech corpus and a special, tonal Punjabi corpus, we ensure that the resulting ASR model is not only accurate but also linguistically robust. Table 4 represents the WER achieved by the model.

Table 4: WER results Before and After LLMs-based GER correction

Dataset used	Before LLM GER correction	After LLM GER Correction (GPT 4.o mini)	After LLM GER Correction (Perplexity)
Hindi	13.63	10.77	10.81
Punjabi	12.94	9.93	9.98

4.5.2 In-context Learning

The efficacy of In-context Learning is evaluated through different experiments. An attention-based encoder-decoder model is used for end-to-end automatic speech recognition task. We utilize GPT-4o mini and perplexity as the correction engine. The main focus is on generated error correction. For these experiments, the 5-best hypotheses are provided as the context input.

To further investigate the model's generative capacity, we conducted a case study on the Hindi dataset (see Table 5). In this instance, the acoustic model struggles with the term अमेज़न. Because अमेज़न was unseen during the training, it was transcribed as phonetically similar but nonsensical tokens like अमेजोन or अमेजन while a traditional re-ranking system would be forced to choose one of these errors, our LLM-based GER successfully inferred the correct token. It achieved this by combining the phonetic signature of the error with the surrounding context.

Table 5: Case study of Incontext learning. The utterance is drawn from Hindi dataset

Type	Utterance	WER (%)
1 st Hypo. By AM	भारत में अमेजोन आर्टिफिशियल इंटेलीजेंस क्षेत्र में निवेश बढ़ा रहा है	10.77
2 nd Hypo. By AM	भारत में अमेजन आर्टिफिशियल इंटेलीजेंस क्षेत्र में निवेश बढ़ा रहा है	9.1
Correction by LLM	भारत में अमेज़न आर्टिफिशियल इंटेलिजेंस क्षेत्र में निवेश बढ़ा रहा है	0
Ground-truth transcription	भारत में अमेज़न आर्टिफिशियल इंटेलिजेंस क्षेत्र में निवेश बढ़ा रहा है।	-

As shown in Table 5, the observation is the introduction of a few in-domain pairs significantly reduces the WER. The model demonstrates a reasoning process where it identifies domain-specific linguistic regulations from the demonstrations and applies

them to the test input. The LLM successfully corrects missing tokens that are entirely absent from the hypotheses list by synthesizing contextual cues.

4.5.3 TAP-based ICT Learning

We extended the evaluation process to TAP-based ICT learning under zero-shot and few shot conditions for both LLMs GPT-4o mini and perplexity. For a baseline comparison, we utilized an attention-encoder-decoder model integrated with pre-trained large language model, same as used in section 4.5.1.

Table 6: WER results by task-activated ICL in zero-shot and few-shot setting. (GPT-4o mini)

Dataset	Baseline	n-shot In-context Learning (ICL)			
		n=0	n=1	n=5	n=10
Hindi	10.77	10.38	10.17	9.51	9.00
Punjabi	9.93	9.55	9.34	8.70	8.23

Table 7: WER results by task-activated ICL in zero-shot and few-shot setting. (Perplexity)

Dataset	Baseline	n-shot In-context Learning (ICL)			
		n=0	n=1	n=5	n=10
Hindi	10.81	10.45	10.25	9.60	9.13
Punjabi	9.88	9.53	9.34	8.69	8.21

The results given in table 6 and 7 indicate that zero-shot performance remains highly sensitive to model scale and task difficulty. On Hindi and Punjabi datasets, the zero-shot pipeline yielded better results, with WER relative reductions of nearly 4%. The emergent zero-shot abilities of LLMs, specifically the capacity to repurpose themselves for error correction without explicit instructions, appear significantly more pronounced. In zero-shot setting, only test hypotheses are available to the LLMs. Direct ICL follow this via task prompts. In few-shot setting, minimal in-domain pairs are available. It contains the hypotheses with ground truth. TAP-based ICL follow this with demonstrations. Our experimental results confirm that both strategies contribute to a reduction in WER. In particular, the few-shot learning solutions, TAP-based ICT with demonstrations allow the LLM to learn the specific mapping between common ASR failure patterns and true transcriptions which provides a robust solution for alleviating the negative impacts of speech variation.

4.5.3 Multi-Pass Augmented Generative Error Correction

The LLM GER provides significant gains in challenging environments but it offers limited improvements when the WER is low [17]. The scope of correction in recognition errors is less with the standard LLM GER. When the ASR error rate is low, LLMs may introduce the errors or hallucinations due to overcorrection which impact the ASR error correction. This limitation is expected to persist in low-resource settings such as Hindi and Punjabi ASR. To address this limitation, multi-pass augmented (MPA) GER approach [47] is proposed as a solution. This approach involves different hypotheses from ASR and LLM models which has the potential to significantly enhance LLM GER. In this, N-best hypotheses are integrated on the input side and apply corrections from LLM on the output side. For MPA GER, the Recognizer Output Voting Error Reduction (ROVER) is used to

combine the results with rescoring. The ROVER is one of the most efficient among traditional methods for reducing speech recognition errors. In this study, MPA GER uses N-best hypotheses combination from single system. The MPA GER utilizes both the advantages of conventional ROVER methods and current LLM approaches. This multi-pass strategy ensures that the correction remains grounded in the acoustic evidence while still benefiting from the linguistic refinement of the LLM. MPA-GER results are compared with baseline and TAP-ICT GER model in table 8.

Table 8: comparison of MPA-GER models' results with Baseline and TAP-ICL GER model

Datasets	Baseline	MPA-GER	TAP-ICL GER (n = 5)
Hindi	10.77	11.34	9.51
Punjabi	9.93	10.83	8.70

The baseline LLM GER error correction may generate extended hallucinations or repetitions. Such hallucinations or repetitions in the LLM GER are sometimes inevitable and are one reason for degrading performance. TAP learning based LLM GER system may generate repetitions, but these repetitions are not included in the GER results. From the results given in table 8, it is observed that TAP learning alleviates hallucinations or repetitions from the text. The proposed model is capable to complement and cancel such hallucinations or repetitions because TAP and ICT learning based LLM process these repetitions differently. As a result, the proposed GER offers higher performance than MPA-GER and the baseline LLM GER.

V. CONCLUSION

In this research, we have presented a robust two-pass E2E ASR framework that effectively bridges the gap between acoustic feature extraction and high-level linguistic reasoning. Using an RNN-T encoder-decoder architecture with an LLM for GER, we have demonstrated a scalable path to ASR that overcome the limitations of traditional Bayesian noisy-channel frameworks. Our main contribution lies in successful transition from resource-intensive fine-tuning to in-context learning and task-activating prompting. These strategies enable pre-trained LLMs to function as sophisticated decoders by using semantic and phonetic priors to resolve recognition errors without the need for weight updates. Our results confirm a fundamental paradigm shift: by leveraging the generative capacity of LLMs, we can recover completely absent tokens from the N-best hypothesis list, effectively surpassing the "oracle WER" upper bound that has long constrained traditional re-ranking methods.

Furthermore, to address the challenges of hallucination and stability in high-accuracy or low-resource languages like Hindi and Punjabi, we compared our model with MPA-GER approach and achieved better results. The proposed framework ensures that the final transcription remains grounded in acoustic reality while leveraging the creative reasoning of generative AI along with deep linguistic refinement. As LLMs continue to scale, their emerging abilities will play an even more important role in ASR pipelines. This work is a sure step towards human-like speech recognition system. It not only transcribes sounds, but also understands context to resolve ambiguity.

Source code: https://drive.google.com/file/d/1Wzav_5a8jn8fPGbwtmqtQOe3cd0m-g64

Conflicts of Interest: Authors declare that there is no conflict of interest for this manuscript.

REFERENCES

1. D. Wang, X. Wang, and S. Lv, "An Overview of End-to-End Automatic Speech Recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.
2. R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-End Speech Recognition: A Survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325-351, 2024.
3. K. Deng et al., "Improving CTC-Based Speech Recognition Via Knowledge Transferring from Pre-Trained Language Models," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8517-8521.
4. Y. Nakagome, T. Komatsu, Y. Fujita, S. Ichimura, and Y. Kida, "InterAug: Augmenting Noisy Intermediate Predictions for CTC-based ASR," in *Interspeech 2022*, 2022, pp. 5140-5144.
5. T. Moriya et al., "Streaming End-to-End Speech Recognition for Hybrid RNN-T/Attention Architecture," in *Interspeech 2021*, Brno, Czechia, 2021.
6. D. Albesano, J. Andrés-Ferrer, N. Ferri, and P. Zhan, "On the Prediction Network Architecture in RNN-T for ASR," in *Interspeech 2022*, Incheon, Korea, 2022.
7. S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4835-4839.
8. S. Ling, G. Ye, R. Zhao, and Y. Gong, "Hybrid Attention-Based Encoder-Decoder Model for Efficient Language Model Adaptation," in *2024 IEEE Spoken Language Technology Workshop (SLT)*, 2024, pp. 49-55.
9. K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 193-199: IEEE.
10. S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Interspeech 2019*, 2019, pp. 1408-1412.
11. Y. C. Chen, S. F. Huang, H. y. Lee, Y. H. Wang, and C. H. Shen, "Audio Word2vec: Sequence-to-Sequence Autoencoding for Unsupervised Learning of Audio Segmentation and Representation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1481-1493, 2019.
12. A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold Fusion: Training Seq2Seq Models Together with Language Models," in *Interspeech 2018*, 2018, pp. 387-391.
13. S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*, 2018, pp. 369-375: IEEE.
14. L. Xu et al., "RescoreBERT: Discriminative Speech Recognition Rescoring With Bert," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6117-6121.
15. H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the Knowledge of BERT for Sequence-to-Sequence ASR," in *Interspeech 2020*, Shanghai, China, 2020.
16. Y. Leng et al., "SoftCorrect: error correction with soft detection for automatic speech recognition," presented at the *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, 2023. Available: <https://doi.org/10.1609/aaai.v37i11.26531>

17. C. Chen, Y. Hu, C. H. H. Yang, S. M. Siniscalchi, P. Y. Chen, and E. S. Chng, "HyPoradise: An Open Baseline for Generative Speech Recognition with Large Language Models," in *Advances in Neural Information Processing Systems*, 2023, pp. 31665-31688: Curran Associates, Inc.
18. C. H. H. Yang, Y. Gu, Y. C. Liu, S. Ghosh, I. Bulyko, and A. Stolcke, "Generative Speech Recognition Error Correction With Large Language Models and Task-Activating Prompting," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1-8.
19. S. Radhakrishnan et al., "Whispering LLaMA: A Cross-Modal Generative Error Correction Framework for Speech Recognition," Singapore, 2023, pp. 10007-10016: Association for Computational Linguistics.
20. E. J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022.
21. J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," presented at the *Proceedings of the 36th International Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2022.
22. D. Kakwani et al., "IndicNLP Suite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages," Online, 2020, pp. 4948-4961: Association for Computational Linguistics.
23. E. Arisoy, A. Sethy, B. Ramabhadran, and S. Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5421-5425.
24. T. Udagawa, M. Suzuki, G. Kurata, N. Itoh, and G. Saon, "Effect and Analysis of Large-scale Language Model Rescoring on Competitive ASR Systems," in *Interspeech 2022*, 2022, pp. 3919-3923.
25. J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, "Masked Language Model Scoring," Online, 2020, pp. 2699-2712: Association for Computational Linguistics.
26. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Minneapolis, Minnesota, 2019, pp. 4171-4186: Association for Computational Linguistics.
27. L. Zhuang, L. Wayne, S. Ya, and Z. Jun, "A Robustly Optimized BERT Pre-training Approach with Post-training," Huhhot, China, 2021, pp. 1218-1227: Chinese Information Processing Society of China.
28. N. M. T. Anh and T. H. Sy, "Improving Speech Recognition with Prompt-based Contextualized ASR and LLM-based Re-predictor," in *Interspeech 2024*, 2024, pp. 737-741.
29. I. E. Kang, C. V. Gysel, and M. H. Siu, "Transformer-based Model for ASR N-Best Rescoring and Rewriting," in *Interspeech 2024*, 2024, pp. 3505-3509.
30. K. Abida, F. Karray, and W. Abida, "A Novel Voting Scheme for ROVER Using Automatic Error Detection," Berlin, Heidelberg, 2012, pp. 175-183: Springer Berlin Heidelberg.
31. K. Audhkhasi, A. M. Zavou, P. G. Georgiou, and S. S. Narayanan, "Theoretical Analysis of Diversity in an Ensemble of Automatic Speech Recognition Systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 3, pp. 711-726, 2014.
32. Y. Fang et al., "Fewer Hallucinations, More Verification: A Three-Stage LLM-Based Framework for ASR Error Correction," *arXiv preprint, arXiv:2505.24347 no. cs.CL*, 2025.
33. H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint, arXiv:2302.13971 no. cs.CL*, 2023.

34. S. Zhang et al., "OPT: Open Pre-trained Transformer Language Models," ArXiv, vol. abs/2205.01068, 2022.
35. A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," presented at the Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA, 2023.
36. S. C. Y. Chan et al., "Data distributional properties drive emergent in-context learning in transformers," presented at the Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 2022.
37. A. Lampinen et al., "Can language models learn from explanations in context?," Abu Dhabi, United Arab Emirates, 2022, pp. 537-563: Association for Computational Linguistics.
38. O. Rubin, J. Herzig, and J. Berant, "Learning To Retrieve Prompts for In-Context Learning," Seattle, United States, 2022, pp. 2655-2671: Association for Computational Linguistics.
39. Y. Zhang, K. Zhou, and Z. Liu, "What Makes Good Examples for Visual In-Context Learning?," in Advances in Neural Information Processing Systems, 2023, pp. 17773-17794: Curran Associates, Inc.
40. S. Lu, I. Bigoulaeva, R. S. Sachdeva, H. Tayyar Madabushi, and I. Gurevych, "Are Emergent Abilities in Large Language Models just In-Context Learning?," vol. 1, L.-W. Ku, A. F. T. Martins, and V. Srikumar, Eds., ed. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 5098-5139.
41. Z. Yang, P. Li, and Y. Liu, "Failures Pave the Way: Enhancing Large Language Models through Tuning-free Rule Accumulation," Singapore, 2023, pp. 1751-1777: Association for Computational Linguistics.
42. R. Ullah et al., "End-to-End Deep Convolutional Recurrent Models for Noise Robust Waveform Speech Enhancement," Sensors, vol. 22, no. 20, p. 7782, 2022.
43. D. Chen and K. Xu, "AReLU: Attention-based Rectified Linear Unit," arXiv preprint arXiv, vol. abs/2006.13858, 2020.
44. T. N. Sainath et al., "Two-Pass End-to-End Speech Recognition," in Proceedings of Interspeech 2019, 2019, pp. 2773-2777.
45. J. K. Sandhu, M. Kumar, and A. Singh, "Punjabi Speech Corpus with Tonal Characteristics," Singapore, 2026, pp. 129-141: Springer Nature Singapore.
46. B. Saglam, X. Hu, Z. Yang, D. Kalogerias, and A. Karbasi, "Learning Task Representations from In-Context Learning," Vienna, Austria, 2025, pp. 6634-6663: Association for Computational Linguistics.
47. Y. Ko, S. Li, C.-H. H. Yang, and T. Kawahara, "Benchmarking Japanese Speech Recognition on ASR-LLM Setups with Multi-Pass Augmented Generative Error Correction," ed, 2024.