Interspeech 2024
1-5 September 2024, Kos, Greece

# Speech Prefix-Tuning with RNNT Loss for Improving LLM Predictions

*Murali Karthick Baskar, Andrew Rosenberg, Bhuvana Ramabhadran, Neeraj Gaur, Zhong Meng*

Google Inc., USA

mkbaskar@google.com,rosenberg@google.com

## Abstract

In this paper, we focus on addressing the constraints faced when applying LLMs to ASR. Recent works utilize prefixLM-type models, which directly apply speech as a prefix to LLMs for ASR. We have found that optimizing speech prefixes leads to better ASR performance and propose applying RNNT loss to perform speech prefix-tuning. This is a simple approach and does not increase the model complexity or alter the inference pipeline. We also propose language-based soft prompting to further improve with frozen LLMs. Empirical analysis on realtime testset from 10 Indic languages demonstrate that our proposed speech prefix-tuning yields improvements with both frozen and fine-tuned LLMs. Our recognition results on an average of 10 Indics show that the proposed prefix-tuning with RNNT loss results in a 12% relative improvement in WER over the baseline with a fine-tuned LLM. Our proposed approches with the frozen LLM leads to a 31% relative improvement over basic soft-prompting prefixLM.

**Index Terms**: speech recognition, large language models, LLM, ASR, prefixLM, prompt-tuning, prefix-tuning

## 1. Introduction

Large language models (LLMs) are revolutionizing automatic speech recognition (ASR) research by addressing various types of prediction errors. PrefixLM is an LLM variant where the input text is accompanied by a prefix. This prefix can take the form of text [1] or speech [2] or image [3] providing additional context for the model. When using speech tokens as prefixes (as in this work), PrefixLM learns to predict text autoregressively, mimicking an end-to-end ASR model [4, 5]. Previous work [6] demonstrates that LLM performance improves with better speech encodings or prefix tokens extracted from self-supervised and supervised models. Scaling the speech encoder also enhances the use of speech prefixes [2, 7], further improving the recognition ability of LLM models such as LLaMA [8]. PrefixLM with speech prefixes have been trained for multiple tasks, including speech recognition and speech translation [9]. Notably, these approaches directly use pretrained LLMs without additional fine-tuning on the target task.

Prefix-tuning [10] offers a lightweight alternative to fine-tuning, as it prepends a trainable token sequence to the text input. Optimizing only the prefix-related parameters adapts the model effectively to downstream tasks. This technique is being incorporated into image and video-based prefixLM models as well. Cross-modal prefix-tuning [11] has been proposed for adapting multilingual translation models to bilingual speech translation tasks. While the final training objective is still only the LLM loss, a pre-trained speech encoder is used for adaptation across modalities. While LLMs gain significant improvement in recognition performance, they still suffer from drawbacks such as higher insertions [12] and code-switching errors [13]. The authors in [12] demonstrate that using LLMs for error correction for ASR helps to improve substitutions compared to RNNTs but increases insertion errors. A simple shallow fusion with a bi-directional LLM leads to code-switching between Mandarin and English when compared to an RNNT model [13]. Despite these studies, our multilingual experiments also indicate that RNNTs do not suffer from insertions and code-switching to the same degree as LLM predictions. We attribute this behavior of RNNT to training with robust aligments and hypothesize that integrating it with LLM will lead to reduced hallucinations and better prediction.

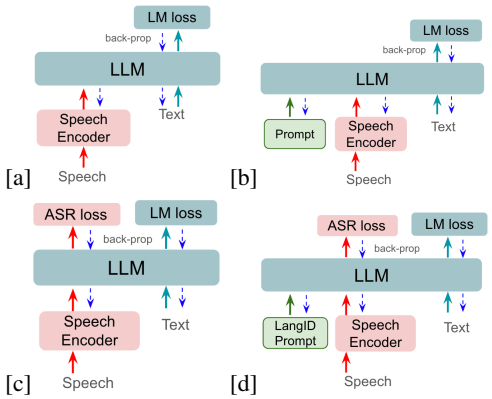



Figure 1: *Previous works [a] and [b] denotes the baseline prefixLM and soft prompting with prefixLM respectively. Subfigures [c] and [d] are our proposed approaches representing the prefix-tuning with RNNT loss and langID based soft prompting respectively*

Our primary contributions are as follows:

- RNNT loss for speech prefix tuning: We demonstrate improvements over both frozen and fine-tuned LLMs. We also examine the constraints encountered when tuning with speech prefixes using Connectionist Temporal Classification (CTC) loss, a non-autoregressive technique. We compare CTC with RNNT loss, highlighting distinctions relevant to prefixLM.
- Language ID (langID) soft prompting: This technique enhances performance of frozen LLMs.
- Bridging the gap: Applying both speech prefix-tuning and langID-based soft prompting can be additive and further reduce the performance gap between frozen and fine-tuned LMs.



10.21437/Interspeech.2024-1903

## 2. Methodology

Our proposed method focuses on finetuning the speech prefix tokens of prefixLM with ASR loss for improved recognition performance. Figures 2 presents the training and evaluation pipeline for our proposed speech prefix-tuning approach. Unlike previous works in figure 1 that solely focus on tuning only the prefix embeddings with the same loss used for text prediction, tuning with RNNT loss updates both speech encoder and prefix embeddings allows the model to learn more discriminant speech features as prefix tokens.

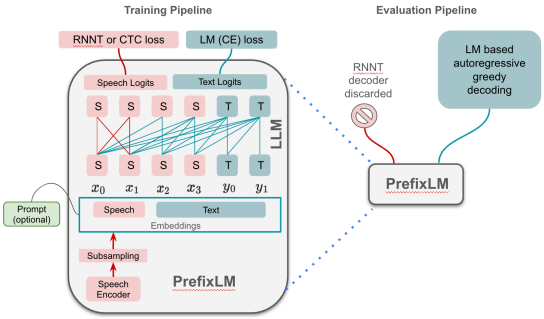



Figure 2: *Training and Evaluation flow for PrefixLM with speech prefix-tuning*

Given a input speech sequence $\mathbf{X} = x_0, x_1, x_2, x_3$ with $T = 4$ frames and text sequence $\mathbf{Y} = y_0, y_1$ with $U = 2$ length, the prefixLM $f(\cdot)$ takes the concatenated input $[X, Y]$.

### 2.1. PrefixLM

PrefixLM [1] is a decoder only model operating in an input-to-target paradigm. It can be viewed as almost encoder-decoder models with shared parameters. PrefixLM has shown competitive advantage [14] as an adaptation method for tasks with relatively small amount of data.The PrefixLM architecture $f(.)$ intakes $[X, Y]$ and enables bi-directional attention on the prefix sequence $x_{0:T-1}$. This serves as the prefix for subsequent prediction on $x_{T:N}$. The output logits $\hat{Z}_{0:T-1} = \hat{x}_{0:T-1}$ corresponding to the acoustic prefix are discarded and the output logits of predicted text $\hat{Z}_{T:T+\hat{U}-1} = \hat{y}_{0:\hat{U}-1}$ are used for decoding. The final training objective to learn the model parameters $\phi$ is done minimizing the errors during next text token prediction using CE loss:

$$\mathcal{L}_{\text{LM}} = -\sum_{u=1}^{\hat{U}} \log p_\phi(\hat{y}_u \mid \hat{y}_{0:u-1}, \mathbf{X}). \tag{1}$$

### 2.2. Prefix-tuning

Prefix-tuning is fine-tuning only the embedding layer which contains the stack of prefixes to guide the text embeddings towards the target task. During training with the equation (1), the prefix embeddings learns the abstract representations of the underlying task. The prefix embeddings remain fixed during inference and projects the required information from the evaluation dataset. Prefix-tuning is computationally efficient with much fewer trainable parameters and also avoids over-fitting.

### 2.3. RNNT decoder

RNNT [15] is an autoregressive sequence-to-sequence model that processes the encoded speech sequence to generate a distribution of text tokens for each timestep of the encoder output. A joint network then combines encoder information and the previous prediction to generate the current token. The RNNT decoder relies on the output of a speech encoder. In our work, we propose using the output logits of a prefixLM's speech prefixes as input to the RNNT decoder.

### 2.4. Speech prefix-tuning with RNNT loss

Based on prior ASR works with LLM [6, 2, 7], we believe that having well encoded speech prefixes act as better context to drive the LM towards the target task. The prefixLM has the ability to learn the speech-to-text alignment better as the speech sequence length gets closer to the text sequence length [2]. Extending this intuition beyond the sequence length, we want to find better speech representations that steers the LM to improve the ASR task. Intuitively, the speech prefixes can influence the text encodings $\mathbf{Y}$ by guiding what to extract from $\mathbf{Y}$; and can improve the text generation by driving the next token distribution. The proposed objective to use ASR loss amplifies the distinctiveness of the speech features by updating the speech related parameters.

To perform speech prefix-tuning, we propagate the speech output logits from LLM to the RNNT decoder. The RNNT loss learns the speech-to-text alignment and $\langle \text{eos} \rangle$ prediction allowing the speech prefixes to accomodate more knowledge of underlying speech and the text to be predicted. The joint training objective is:

$$\mathcal{L}_{\text{RNNT}} = -\log p(\mathbf{Y} \mid \hat{\mathbf{X}}) \tag{2}$$

$$\mathcal{L}_{\text{joint}} = \alpha \cdot \mathcal{L}_{\text{LM}} + (1 - \alpha) \cdot \mathcal{L}_{\text{RNNT}} \tag{3}$$

Here, $\hat{\mathbf{X}}$ is the speech prefix output (speech logits) from the LLM as in figure 2.

### 2.5. Language ID based soft prompting

Prefix tuning learns task specific information and conditioning the prefix with LangID allows it to learn language specific embeddings. This helps to stabilize performance on multiple languages when the LLM is frozen. Figure 1[d] shows the training and eval pipeline for langID based prompting. The LangID embeddings are of size $[L \times M \times D]$, where $L$ is the number of languages, $M$ is the prompt length and $D$ is the number of dimensions. The soft prompt $[M \times D]$ is chosen corresponding to the langID from the source input and is fed along with the speech prefixes from Section 2.4. During training, the soft prompt embeddings are only updated and are fixed during inference.

### 2.6. LLM training

LLMs can be trained in one of the following ways when using speech prefixes:

- Frozen LLM: The RNNT loss updates only the speech encoder and the soft prompt embeddings while the entire LLM parameters $\phi$ are kept frozen. The LM loss (1) updates the soft prompt embeddings only.
- Finetuned LLM: LLM parameters are updated simultaneously with both ASR and LM loss. The soft prompt embeddings are tuned only with the LM loss given in equation (1).

Table 1: *WER on all 10 Indic languages using prompt tuning, prefix tuning and language ID prompt tuning with and without fine-tuning the LLM.*

| LLM | Tune | bn | en | gu | hi | kn | ml | mr | ta | te | ur | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| USM-L + frozen LLM-L | - | 33.5 | 16.6 | 51.1 | 49.4 | 57.6 | 54.6 | 30.3 | 52.2 | 45.7 | 45.7 | 43.6 |
| | Prompt | 33.5 | 15.2 | 50.2 | 45.1 | 52.3 | 51.1 | 30.4 | 52.0 | 44.6 | 41.0 | 41.5 |
| | Prefix | 22.0 | 14.1 | 37.8 | 15.5 | 37.6 | 40.1 | 27.3 | 42.0 | 33.0 | 21.4 | 29.1 |
| | Prefix+Prompt | 20.9 | 14.5 | 37.6 | 15.3 | 37.4 | 39.7 | 26.9 | 42.2 | 32.7 | 21.3 | 28.9 |
| | Prefix+LangIDPrompt | 20.5 | 14.3 | 37.0 | 15.2 | 37.2 | 39.4 | 26.4 | 41.1 | 32.4 | 21.0 | 28.5 |
| USM-L + finetuned LLM-L | - | 27.5 | 17.4 | 40.7 | 18.3 | 40.6 | 42.8 | 29.9 | 44.0 | 35.6 | 25.6 | 32.2 |
| | Prefix | 20.2 | 13.7 | 37.1 | 15.2 | 37.2 | 39.5 | 26.5 | 41.5 | 32.4 | 21.0 | 28.3 |

# 3. Experiments

## 3.1. Datasets and Models

*Data*: The training data used in these experiments is composed of YouTube longform data as described in [16, 17] and drawn from 10 Indic languages. All data is drawn from 10 Indic languages and segmented into "utterances" with a maximum length of 30s. Language information is obtained from the uploaded language tag in the video and incorporated as an auxiliary embedding along with speech features. For evaluation we use a YouTube test set for 10 Indic languages which combines utterances spanning a broad set of topics ranging from sports and entertainment to education. Both training and test data is segmented into utterances with a maximum length of 30s. See Table 2 for the distribution of training and test material across languages.

Table 2: *Training and testing data statistics*

| LID | Language | #Hours (Train) | #Hours (Test) |
|---|---|---|---|
| bn | Bengali | 3.3k | 30.2 |
| en | English | 3.5k | 22.2 |
| gu | Gujarati | 3.5k | 30.4 |
| hi | Hindi | 5.5k | 30.1 |
| kn | Kannada | 3.6k | 29.8 |
| ml | Malayalam | 3.2k | 29.3 |
| mr | Marathi | 3.7k | 30.0 |
| ta | Tamil | 4.5k | 28.7 |
| te | Telugu | 4.2k | 29.6 |
| ur | Urdu | 2.0k | 30.2 |

*Speech Encoder*: We employ universal speech models (USM) [18] with model complexity of 300M (USM-S) and 600M (USM-L) parameters. USM-S leverages a 24-layer Conformer with a model dimension (768) resulting in a total of 333.5 million parameters while USM-L has the same number of layers as USM-S but with 1024 dimensions. Both USM architectures use chunk-wise bi-directional attention allowing them to accurately model long audio sequences (30-second segments during training). Mel fiterbank based speech features are fed to the USM speech encoder and the encoded outputs are subsampled by factor of 4 (160ms frame rate) for efficiency. This subsampled encoder output $\mathbf{X}$ serves as the prefix embedding. The USM is trained using a large amount of multilingual data: over 10 million hours of unlabeled audio, tens of billions of text sentences, over a hundred thousand hours of supervised and semi-supervised audio. The data is drawn from over a hundred languages covering various topics [18].

*LLM*: The large language model used in this paper builds upon the JAX based M4 multipod model [19]). This a Transformer based decoder only model. In this paper, we present results with two LLM sizes, 128M and 500M parameters. 128M has 8 layers with 16 heads, 4096 hidden dimensions. 500M model has 30 layers with 16 heads and 4096 hidden dimensions. The feed-forward layer configuration is common to both 128M (LLM-S) and 500M (LLM-L) parameter models with 16384 dimensions and the attention head size is 64. Both these models are trained with 800B text tokens. We use relative positional embeddings and GELU activations. Adafactor optimizer with momentum is used for training with a batch size of 1024 and a sequence lengths of 1k tokens. Finally, the model is quantized to bfloat16 precision for efficient tuning and inference. 256k vocab based sentencepiece tokens [20] are used for training. Training is performed using a variant of UL2 objective [21], as described in [22].

# 4. Results and Discussion

## 4.1. PrefixLM with CTC and RNNT

The introduction of RNNT or CTC decoders and corresponding losses to a PrefixLM ASR model demonstrates clear improvements. Using a CTC auxiliary loss, WER on USM-L drops from 32.2 to 31.8 with a larger win on USM-S, where performance goes from 36.7 to 33.7 (8% relative). RNNT in isolation is a better ASR model than CTC (29.4 vs. 33.0) given by rows 1 and 5 in Table 3. This is also reflected in its combination with PrefixLM. The introduction of the RNNT decoder and loss to PrefixLM yields 5.4 % and 3.7% relative wins on USM-S and USM-L respectively.

Table 3: *WER on the average of 10 Indic languages using CTC, RNNT and PrefixLM using 300M (USM-S) and 600M (USM-L) speech encoders. PrefixLM (finetuned) model is trained using $\mathcal{L}_{\text{LM}}$ in* (1) *and PrefixLM (prefix-tuned with RNNT) model is trained using $\mathcal{L}_{\text{joint}}$ in* (3)*.*

| Train decoder | Eval decoder | Avg WER (%) | |
|---|---|---|---|
| | | USM-S | USM-L |
| CTC | CTC | 35.9 | 33.0 |
| PrefixLM (finetuned) | LM | 36.7 | 32.2 |
| PrefixLM (prefix-tuned with CTC) | CTC | 33.8 | 31.8 |
| PrefixLM (prefix-tuned with CTC) | LM | 33.7 | 31.8 |
| RNNT | RNNT | 31.5 | 29.4 |
| PrefixLM (prefix-tuned with RNNT) | RNNT | 29.8 | 28.4 |
| PrefixLM (prefix-tuned with RNNT) | LM | 29.8 | 28.3 |

## 4.2. Language-based Prompting allows the LM to be frozen

The results in Table 3 update the full PrefixLM model (Figure 2), both the speech encoder and LM. However, updating the LLM is computationally expensive. In this section we explore in-context, prompting techniques to achieve similar per-

formance by updating the speech encoder while keeping the LLM frozen.

Table 1 also shows the results of using both learned soft-prompts and language-conditioned prompts as described in Section 2.5 using LLM-L with USM-L as the speech encoder. The average %WER performance of using frozen LLM is significantly worse with 43.6 over finetuned LLM with 28.3. Prompt tuning shows that it improves on average by absolute 2.1%. Our proposed speech prefix-tuning brings the average %WER down to 29.1. Tuning both speech prefixes and prompt embeddings is complimentary and shows marginal gain. We hypothesize this marginal improvement with speech prefix+prompt tuning is due to the limitation of using a single prompt embedding to model the multilingual input data. Extending the prompt to be language specific by using the langID prompt tuning is able to bring the performance of the frozen LLM to within 0.2% WER of the best performing fully-updated model in Table 1 (28.5 vs. 28.3). This results in a model where approximately half of the parameters do not need to be updated with a very modest impact on quality.

### 4.3. Error analysis of speech prefix-tuning

| Lang | Error | RNNT | PrefixLM (finetuned) | PrefixLM (prefix-tuned with RNNT) |
|---|---|---|---|---|
| bn | D | 3.6 | 3.6 | 4.6 |
| | I | 2.0 | 6.7 | 1.9 |
| | S | 14.6 | 17.2 | 13.6 |
| en | D | 3.1 | 3.0 | 3.6 |
| | I | 2.7 | 5.4 | 2.3 |
| | S | 11.6 | 9.0 | 7.8 |
| gu | D | 5.4 | 5.5 | 5.5 |
| | I | 8.8 | 11.7 | 8.6 |
| | S | 23.7 | 23.5 | 23.0 |
| hi | D | 3.8 | 3.5 | 3.1 |
| | I | 2.2 | 3.8 | 2.1 |
| | S | 23.7 | 11.0 | 10.0 |
| kn | D | 5.8 | 5.7 | 5.6 |
| | I | 4.1 | 6.7 | 4.3 |
| | S | 27.8 | 28.2 | 27.3 |
| ml | D | 5.3 | 6.1 | 5.3 |
| | I | 7.4 | 9.3 | 7.2 |
| | S | 27.2 | 27.5 | 26.1 |
| mr | D | 5.4 | 6.0 | 6.5 |
| | I | 2.6 | 5.5 | 2.4 |
| | S | 18.7 | 18.4 | 17.6 |
| ta | D | 6.1 | 5.4 | 5.5 |
| | I | 5.5 | 7.6 | 5.7 |
| | S | 31.0 | 31.1 | 30.3 |
| te | D | 4.5 | 4.4 | 4.7 |
| | I | 5.4 | 7.7 | 5.4 |
| | S | 23.0 | 23.5 | 22.3 |
| ur | D | 3.4 | 4.0 | 3.1 |
| | I | 6.2 | 7.6 | 6.3 |
| | S | 19.4 | 13.9 | 11.7 |

Table 4: **D***eletion* / **I***nsertion* / **S***ubstitution rate across PrefixLM and RNNT for Indian languages. The errors are color coded as higher errors and lower errors between the finetuned prefixlm $\mathcal{L}_{\text{LM}}$ as in* (1) *and PrefixLM with speech prefix-tuning ($\mathcal{L}_{\text{joint}}$) in* (3).

Table 4 shows that the PrefixLM (finetuned) model demonstrates a substantially higher rate of insertions.This is due to the effect of hallucinations during decoding. Our proposed approach speech prefix-tuning with RNNT using $\mathbf{L}_{\text{joint}}$ loss reduces this insertion rate while maintaining the overall quality. The average performance gains in table 3 are attributed primarily to improvement in insertions and substitutions without hurting the deletion rate. We observe this behavior for all 10 Indic languages.

### 4.4. Code-switching analysis

Without language ID information, multilingual ASR models have a tendency to produce hypotheses in multiple languages, sometimes multiple scripts. Some of this is by design, as speech in Indian languages is frequently code mixed. However, this is also a source of error where a hypothesis may be acoustically "correct" but produced in an unexpected script. Here we measure the code-mixing behavior of the different approaches using the Code Mixing Index (CMI) measure denoted in [23]:

$$\text{CMI} = \begin{cases} 100 * [\frac{1-\max(w_i)}{n-u}]; & n > u \\ 0 \qquad\qquad\qquad\quad ; & n = u \end{cases}$$

Here, $\max(w_i)$ = highest number of words present from any language (more than 1 language can have the same highest word count), $n$ = no of tokens in utterance $x$, $u$ = number of tokens given other language tags.

Table 5: *Code Mixing percentage on Tamil (ta) for finetuned prefixlm $\mathcal{L}_{\text{LM}}$ and PrefixLM with speech prefix-tuning ($\mathcal{L}_{\text{joint}}$) models*

| Model | # words | # maxwords in ta | CMI |
|---|---|---|---|
| RNNT | 137257 | 136788 | 34.2% |
| PrefixLM (finetuned) | 137394 | 136888 | 36.8% |
| Proposed | 137299 | 136849 | 32.8% |

In Table 5 we see that, for Tamil (ta), the PrefixLM generates more code-mixed hypotheses than the proposed speech prefix-tuning $\mathcal{L}_{\text{joint}}$ model. The %CMI improves from 36.8 to 32.8 which shows that the number of non Tamil words are predicted less compared to the baseline. This behavior was observed across other Indic languages as well.

## 5. Related works

In just the last few years there has been a lot of work on using LLMs within ASR. These include Flamingo [24], PrefixLM [1], and SLM [9]. Some other notable works [25, 26, 27], merge pretrained speech encoder with pretrained text based LLM to perform ASR and other speech related tasks. Given the rate of progress this is a necessarily incomplete list. Most of these works rely on a well pretrained speech encoder [18] with matching domain to achieve better recognition performance. On the other hand, the LLMs are adapted to the target domain by performing either complete finetuning [28, 29] or other lightweight approaches such as prompt-tuning [30, 6, 31] .

In this work, we show that these two techniques can be unified by performing speech prefix-tuning using a joint RNNT and LM loss provides both better prefix embeddings and also performs lightweight finetuning. To further reduce the tunable parameters, we present the langID based soft prompting in Section 2.5 by using conditioning information to learn a soft-prompt, rather than hand crafting a hard-prompt for this conditioning.

## 6. Conclusions

The inclusion of traditional RNNT loss successfully complements the success of PrefixLM-based ASR. The overall quality remains high, while balancing the insertion rate. Moreover the rate of code-mixed output is reduced. We have also demonstrated the value of learned soft-prompts conditioned on language ID as a route to eliminate the need for a fine-tuned LM, substantially reducing the training required for this technique to obtain high quality results.

# 7. References


[1] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[2] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli *et al.*, "Prompting large language models with speech recognition abilities," *arXiv preprint arXiv:2307.11795*, 2023.

[3] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[4] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiát, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 521–527.

[5] M. Kim, K. Sung-Bin, and T.-H. Oh, "Prefix tuning for automated audio captioning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[6] Z. Ma, G. Yang, Y. Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang, and X. Chen, "An Embarrassingly Simple Approach for LLM with Strong ASR Capacity," *arXiv e-prints*, p. arXiv:2402.08846, Feb. 2024.

[7] E. Lakomkin, C. Wu, Y. Fathullah, O. Kalinli, M. L. Seltzer, and C. Fuegen, "End-to-end speech recognition contextualization with large language models," *arXiv preprint arXiv:2309.10917*, 2023.

[8] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[9] M. Wang, W. Han, I. Shafran, Z. Wu, C.-C. Chiu, Y. Cao, N. Chen, Y. Zhang, H. Soltau, P. K. Rubenstein *et al.*, "Slm: Bridge the thin gap between speech and text foundation models," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[10] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.

[11] Y. Ma, T. H. Nguyen, and B. Ma, "Cpt: Cross-modal prefix-tuning for speech-to-text translation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6217–6221.

[12] R. Ma, M. Qian, P. Manakul, M. Gales, and K. Knill, "Can generative large language models perform asr error correction?" *arXiv preprint arXiv:2307.04172*, 2023.

[13] K. Hu, T. N. Sainath, B. Li, N. Du, Y. Huang, A. M. Dai, Y. Zhang, R. Cabrera, Z. Chen, and T. Strohman, "Massively multilingual shallow fusion with large language models," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] N. Ding, T. Levinboim, J. Wu, S. Goodman, and R. Soricut, "Causallm is not optimal for in-context learning," in *The Twelfth International Conference on Learning Representations*, 2023.

[15] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7049–7053.

[16] H. Liao, E. McDermott, and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 368–373.

[17] T. Chen, C. Allauzen, Y. Huang, D. Park, D. Rybach, W. R. Huang, R. Cabrera, K. Audhkhasi, B. Ramabhadran, P. J. Moreno, and M. Riley, "Large-scale language model rescoring on long-form data," 2023.

[18] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang *et al.*, "Google usm: Scaling automatic speech recognition beyond 100 languages," *arXiv e-prints*, pp. arXiv–2303, 2023.

[19] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[20] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 66–71.

[21] Y. Tay, M. Dehghani, V. Q. Tran, X. Garcia, D. Bahri, T. Schuster, H. S. Zheng, N. Houlsby, and D. Metzler, "Unifying language learning paradigms," *arXiv e-prints*, pp. arXiv–2205, 2022.

[22] X. Garcia, Y. Bansal, C. Cherry, G. Foster, M. Krikun, M. Johnson, and O. Firat, "The unreasonable effectiveness of few-shot learning for machine translation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 867–10 878.

[23] B. Gambäck and A. Das, "On measuring the complexity of code-mixing," in *Proceedings of the 11th international conference on natural language processing, Goa, India*, 2014, pp. 1–7.

[24] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.

[25] Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng *et al.*, "Lauragpt: Listen, attend, understand, and regenerate audio with gpt," *arXiv preprint arXiv:2310.04673*, 2023.

[26] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," *arXiv preprint arXiv:2305.11000*, 2023.

[27] W. Yu, C. Tang, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "Connecting speech encoder and large language model for asr," *arXiv preprint arXiv:2309.13963*, 2023.

[28] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu *et al.*, "On decoder-only architecture for speech-to-text and large language model integration," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[29] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, M. Zejun, and C. Zhang, "Salmonn: Towards generic hearing abilities for large language models," in *The Twelfth International Conference on Learning Representations*, 2023.

[30] H. Yu, H. Zheng, Y. Zhang, S. Xie, X. Cao, and Z. Fang, "Prompt tuning is all we need?" 2024. [Online]. Available: https://openreview.net/forum?id=eBTtShIjxu

[31] X. Liu, K. Ji, Y. Fu, Z. Du, Z. Yang, and J. Tang, "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," *CoRR*, vol. abs/2110.07602, 2021. [Online]. Available: https://arxiv.org/abs/2110.07602