Hot-Fixing Wake Work Recognition for End-to-End ASR via Neural Model Reprogramming

Pin-Jui Ku[*], ***I-Fan Chen***, *Chao-Han Huck Yang, Anirudh Raju, Pranav Dheram, Pegah Ghahremani, Brian King, Jing Liu, Roger Ren, Phani Sankar Nidadavolu*

[*] Georgia Tech, Georgia, USA [2] Amazon Alexa AI, USA

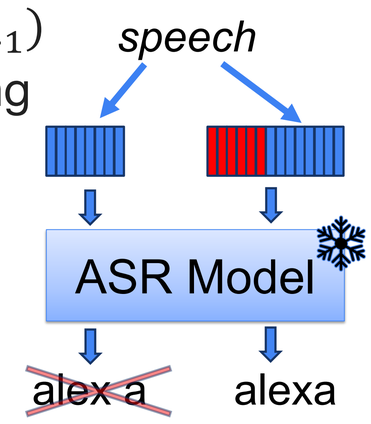
ICASSP 2024 KOREA

Introduction

- **Motivation**: Hotfix the deployed ASR model without updating the model weights.
- **Contributions**
  - Proposed two neural reprogramming approaches for RNN-T based ASR models.
  - Verify the effectiveness of the approach on wake word recognition tasks.
  - In depth analyses show the advantage and limitation of the approaches

Methodology

- **Problem Formulation**
  - Fix incorrect output $y_i^e = f(\mathbf{X}_t, \mathbf{Y}_{i-1})$ from streaming ASR by introducing $g()$, where
    - $\widetilde{\mathbf{X}}_t, \widetilde{\mathbf{Y}}_{i-1} = g(\mathbf{X}_t, \mathbf{Y}_{i-1})$
  - so that $y_i = f\big(g(\mathbf{X}_t, \mathbf{Y}_{i-1})\big)$



- **Approach 1: Trigger Frames**
  - $g(\mathbf{X}_t, \mathbf{Y}_{i-1}) = [\mathbf{T}; \mathbf{X}_t], \mathbf{Y}_{i-1}$
    - **T**: Trainable prepending feature frames
- **Approach 2: RNN-T Predictor-State Initialization**
  - $g(\mathbf{X}_t, \mathbf{Y}_{i-1}) = \mathbf{X}_t, [\mathbf{T}; \mathbf{Y}_{i-1}]$
  - Equivalent to the customized prediction-state initialization for any stateful prediction network model.

Experimental Setup

- **Scenario**
  - Adapting the 76M **LibriSpeech** pretrained torch audio Emformer RNN-T model (B1) to recognize voice command speech w/ wake words.

System	# Trainable Param inside ASR Model	# Trainable Param outside ASR Model
B2: Finetuning	76 M	0
E1: Trigger-Frame	0	3,200
E2: Pred-State-Init	0	3,072

- **Data**
  - Synthesized voice command speech w/ wake words
  - [5 wake words + SLURP sentences] + ESPNet TTS
  - Wake words: ***Alexa***, ***Cortana***, ***Disney***, ***Google***, ***Siri***
  - Example utterances
    - w/o WW: "**send a request to Martin**"
    - WW: "***Cortana* send a request to Martin**"
  - The following data was used for each wake word:
    - 60 x 2 = 120 training utts; 1049 x 2 = 2098 val utts
    - 1524 x 2 (spkr) = 3048 eval utt
  - Also evaluated on LibriSpeech Test Clean

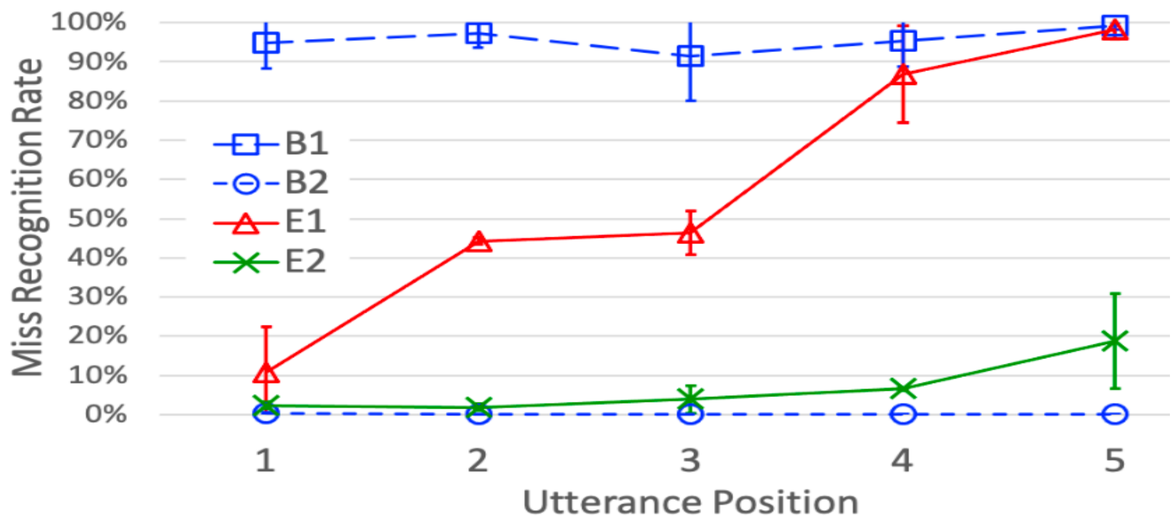
Results

- The averaged False Rejection Rate (FRR) over the five wake words and Word Error Rate (WER) on the synthesized WW and w/o WW voice command eval utterances and LibriSpeech test-clean datasets

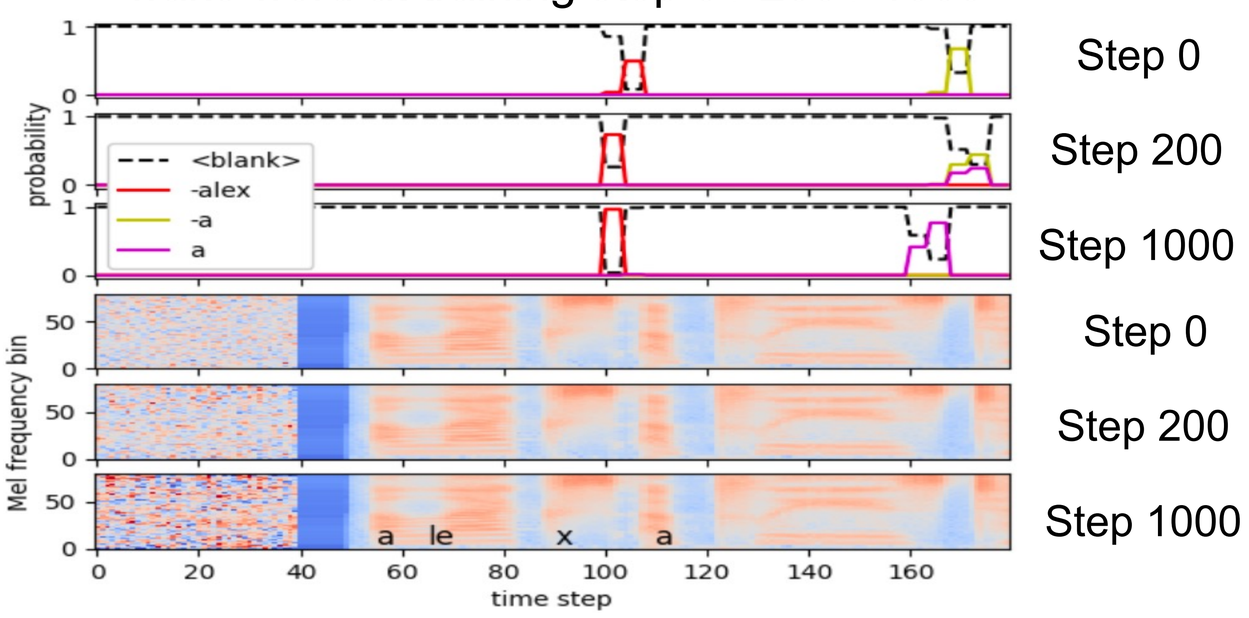
System	FRR (%)	WER (%)		
		WW	w/o WW	Libri
B1: pretrained	98.1	27.0	7.8	4.6
B2: finetuning	0.1	4.0	4.8	4.7
E1: trigger-frame	22.9	11.5	8.9	4.8
E2: pred-state init	2.8	7.0	8.7	4.7

- **Analysis: Impact of Target Word Utterance Position**
  - Place the target wake-word at utterance position I following the template.
  - Effectiveness of the approach reduced as the utt position of the target word increased.

Position	Template
1	<wake_word_name> ...
2	Call <wake_word_name> ...
3	Tell me <wake_word_name> ...
4	How are you <wake_word_name> ...
5	Do me a favor <wake_word_name> ...



- **Visualization of the trigger frame learning process**
  - Trigger frames and ASR outputs for the "***Alexa***" wake word at training step 0 / 200 / 1000



Conclusion

- We can effectively hotfix the ASR models without updating the model weights.
- The effectiveness of the current approaches suffers from the distance to the reprogramming injection place, which can be a future research direction.