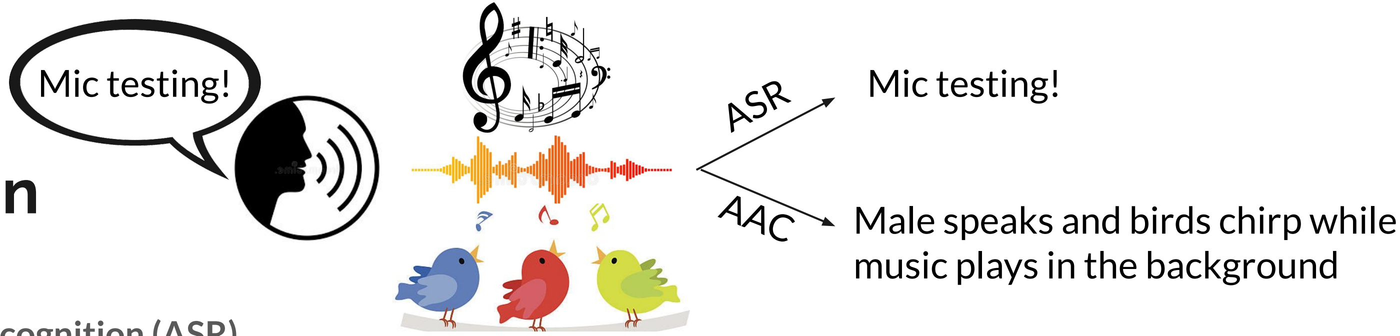
ICASSP 2022 SPE-65: Speech Recognition: General Topics IV

Joint Speech Recognition & Audio Captioning

Chaitanya Narisetty[*], Emiru Tsunoo[†], Xuankai Chang[*], Yosuke Kashiwagi[†], Michael Hentschel[†], Shinji Watanabe[*]

[*] WAVLab, Carnegie Mellon University
[†] R&D Center, Sony Group Corporation

Introduction



Mic testing!

ASR

Mic testing!

AAC

Male speaks and birds chirp while music plays in the background

Automatic Speech Recognition (ASR)

- End-to-end ASR is a seq-to-seq task that transcribes human speech
- Real-world speech samples are often contaminated with background noise(s)

Automated Audio Captioning (AAC)

- Generate natural language descriptions of contents in audio samples
- AAC, together with ASR, provides a holistic understanding and better interpretability of audio
- Such integration can improve the video viewing experience for the hearing impaired

Overview

Motivation

- The ASR model's ability to adapt to various background sounds determines its performance in noisy environments. This **makes both ASR and AAC tasks interdependent**
- **Both tasks are audio-to-text generation tasks**

Challenge: lack of an audio dataset containing both transcription and caption labels

Contribution

- Present the **first attempt to jointly model ASR and AAC** as a multi-task problem
- **Prepare a synthetic multi-task dataset:** combine clean speech and captioned non-speech samples
- **Propose joint modeling approaches** that outperform independently trained models

Multi-task Dataset Synthesis

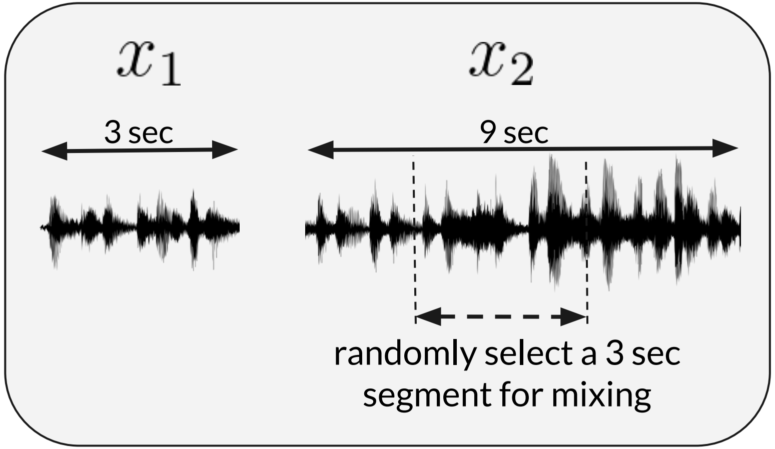
- Synthetically **mix transcribed clean speech with captioned non-speech samples**
- Datasets for ASR: *WSJ*, *LibriSpeech* etc. and for audio captioning: *AudioCaps*, *Clotho* etc.
- In this work, we **mix *WSJ* (37k samples) and *AudioCaps* (46k samples from YouTube)**
- Notes:
  - **Captioned audio may contain speech, so remove them** (~20k samples)
  - **Look for substrings “speak” or “talk”** (takes care of “speaks”, “speaking”, “talks”, “talking” etc.) in an audio caption (e.g. “adult male begins **speak**ing followed by muffled sound”)

Audio Mixing Process

- Randomly pick a clean speech x_1 and a non-speech x_2
- Normalize their amplitudes to [-1 1] range to get $\hat{x}_1$ and $\hat{x}_2$
- Mix using a scalar mixing weight: $\gamma \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$

$$x_{\text{mix}} = \hat{x}_1 + \gamma \cdot \hat{x}_2$$

- Here, γ controls the amount of background sounds

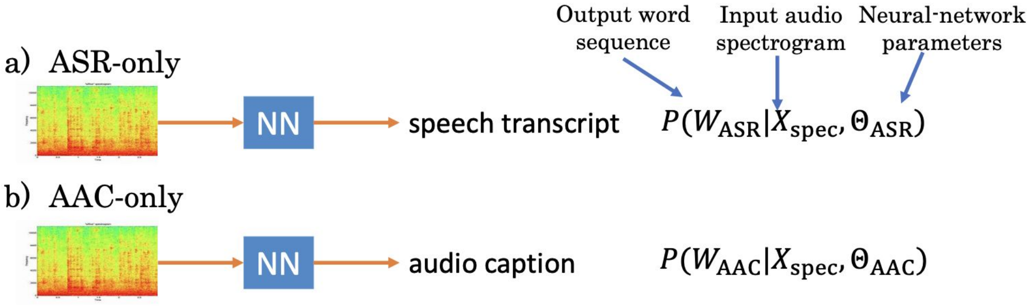


randomly select a 3 sec segment for mixing

🔊 🔊 🔊 🔊 🔊 🔊 🔊

$\hat{x}_1$ $\hat{x}_2$ γ: 0.1 0.2 0.4 0.6 0.8

Independent Modeling


Output word sequence
Input audio spectrogram
Neural-network parameters
a) ASR-only
NN
speech transcript
$P(W_{\mathrm{ASR}}|X_{\mathrm{spec}}, \Theta_{\mathrm{ASR}})$
b) AAC-only
NN
audio caption
$P(W_{\mathrm{AAC}}|X_{\mathrm{spec}}, \Theta_{\mathrm{AAC}})$

ASR-only

- Typically a **Transformer based encoder-decoder** framework (recently based on RNN Transducers)
- Optimized **using both CTC loss and attention loss**
- Integrated with pretrained language models during inference (but not considered in this work)

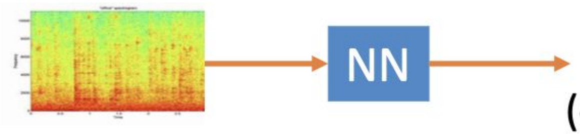
AAC-only

- Also based on a **Transformer based encoder-decoder** framework
- **Cannot use CTC loss** due to lack of temporal alignment between audio and its caption

Proposed Joint Modeling Approaches

1. Cat-ASR-AAC and Cat-AAC-ASR: concatenate ASR and AAC word sequences

$$P(W_{\text{ASR}}, W_{\text{AAC}} | X_{\text{spec}}, \Theta_{\text{Cat}})$$

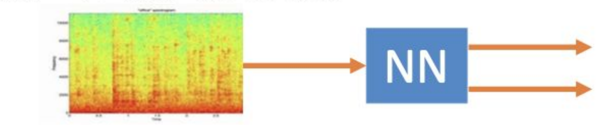


speech transcript *<sep. token>* audio caption

(or) audio caption *<sep. token>* speech transcript

2. Dual-decoder: have a separate decoder for ASR and AAC word sequences

$$P(W_{\text{ASR}} | X_{\text{spec}}, \Theta_{\text{DD}}) \cdot P(W_{\text{AAC}} | X_{\text{spec}}, \Theta_{\text{DD}})$$



speech transcript

audio caption

Concatenating Output Sequences

- Similar to previously studied serialized output training [Kanda, N., Gaur, et. al. 2020]
- Con**cat**enate word sequences of ASR and AAC using a separation token

$$W_{\text{Cat}} = \text{Concat}(\ W_{\text{ASR}}, \langle \textit{sep. token} \rangle\ , W_{\text{AAC}}\ )$$

$$P(W_{\text{Cat}}|X_{\text{spec}}, \Theta_{\text{Cat}}) = P(W_{\text{ASR}}|X_{\text{spec}}, \Theta_{\text{Cat}}) \cdot P(W_{\text{AAC}}|W_{\text{ASR}}, X_{\text{spec}}, \Theta_{\text{Cat}})$$

- Concatenate word sequences in **rev**erse order

$$W_{\text{Rev}} = \text{Concat}(\ W_{\text{AAC}}, \langle \textit{sep. token} \rangle\ , W_{\text{ASR}}\ )$$

$$P(W_{\text{Rev}}|X_{\text{spec}}, \Theta_{\text{Rev}}) = P(W_{\text{AAC}}|X_{\text{spec}}, \Theta_{\text{Rev}}) \cdot P(W_{\text{ASR}}|W_{\text{AAC}}, X_{\text{spec}}, \Theta_{\text{Rev}})$$

Kanda, N., Gaur, Y., Wang, X., Meng, Z. and Yoshioka, T., 2020. Serialized output training for end-to-end overlapped speech recognition. INTERSPEECH 2020

Concatenating Output Sequences: Drawbacks

- Inference may be difficult because of **increased length of output sequence**
- **Cannot use CTC loss** due to loss of temporal alignment
- Decoder learns distribution of ASR and AAC token transitions, but they need not be correlated

Dual Output Decoding

- Have a separate decoder for each of ASR and AAC word sequences
- Assumes conditional independence given input features and model parameters

$$P(W_{\mathrm{ASR}}, W_{\mathrm{AAC}}|X_{\mathrm{spec}}, \Theta_{\mathrm{DD}}) = P(W_{\mathrm{ASR}}|X_{\mathrm{spec}}, \Theta_{\mathrm{DD}}) \cdot P(W_{\mathrm{AAC}}|X_{\mathrm{spec}}, \Theta_{\mathrm{DD}})$$

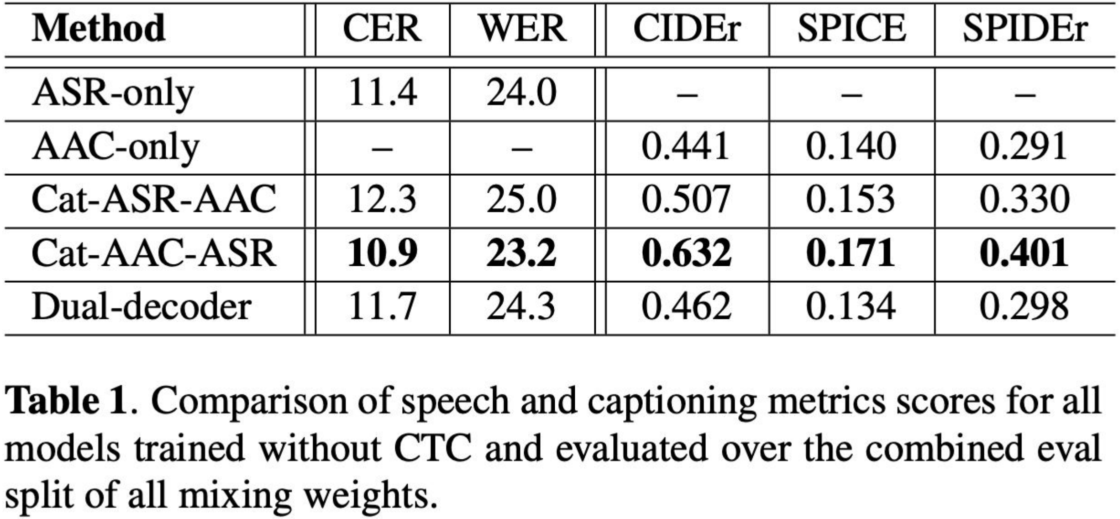
- Allows use of the CTC loss to be integrated (linearly interpolated) with attention loss

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{\mathrm{ctc}} + (1 - \lambda) \cdot \mathcal{L}_{\mathrm{att}}$$

Experiments

- Synthetic dataset is obtained by mixing 37k *WSJ* samples and 26k non-speech *AudioCaps* samples
- Five choices for scalar mixing weight: 37k x 5 = **187k total training samples**
- Dev+Eval sets of *WSJ*: ~800 samples are mixed with an unseen set of *AudioCaps*: ~600 samples
- Independent and joint models are trained on entire training set, and tested for each mixing weight
- **Architecture: Transformer** encoder (12 layers) and decoder (6 layers) with 4 attention heads
- ASR metrics: **CER, WER** (lower is better)
- AAC metrics: **CIDEr, SPICE, SPIDEr** (higher is better)

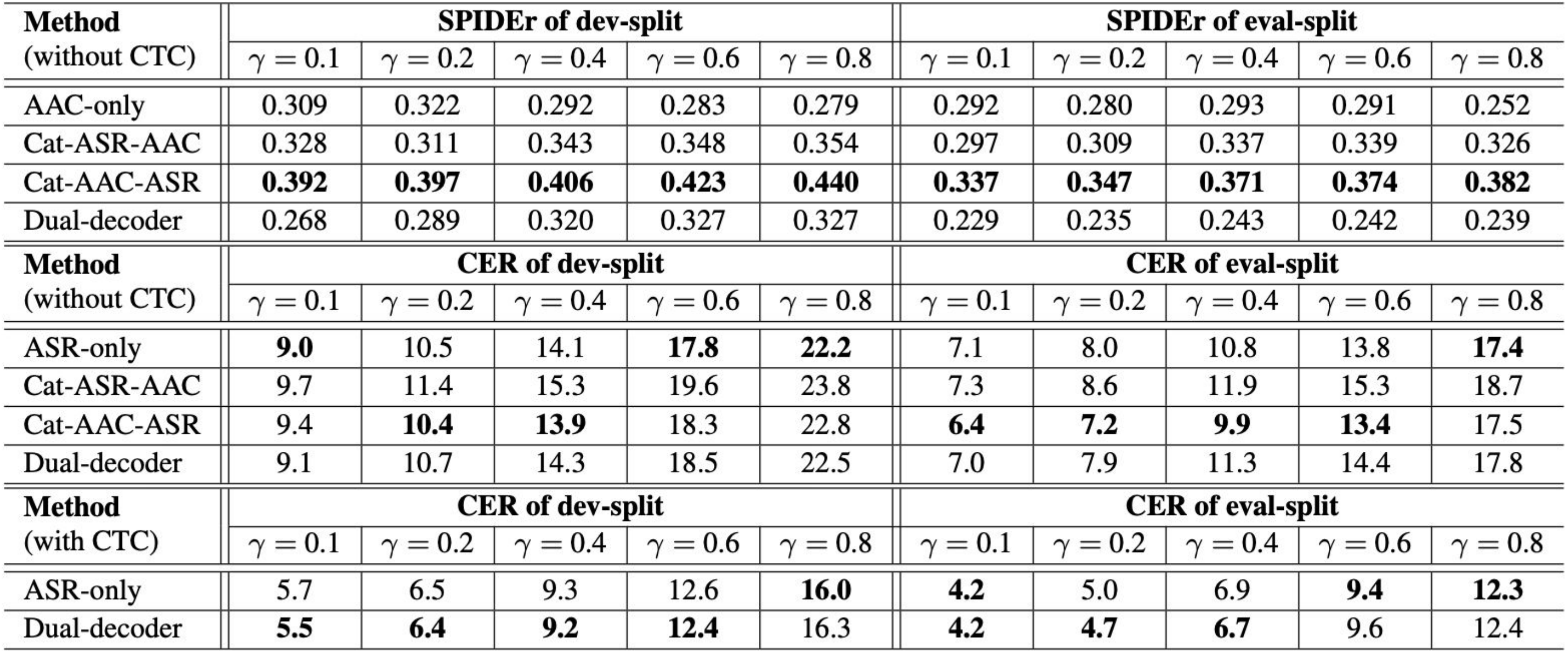
Overall Performance Without CTC on Eval-split



Method	CER	WER	CIDEr	SPICE	SPIDEr
ASR-only	11.4	24.0	–	–	–
AAC-only	–	–	0.441	0.140	0.291
Cat-ASR-AAC	12.3	25.0	0.507	0.153	0.330
Cat-AAC-ASR	**10.9**	**23.2**	**0.632**	**0.171**	**0.401**
Dual-decoder	11.7	24.3	0.462	0.134	0.298

Table 1. Comparison of speech and captioning metrics scores for all models trained without CTC and evaluated over the combined eval split of all mixing weights.

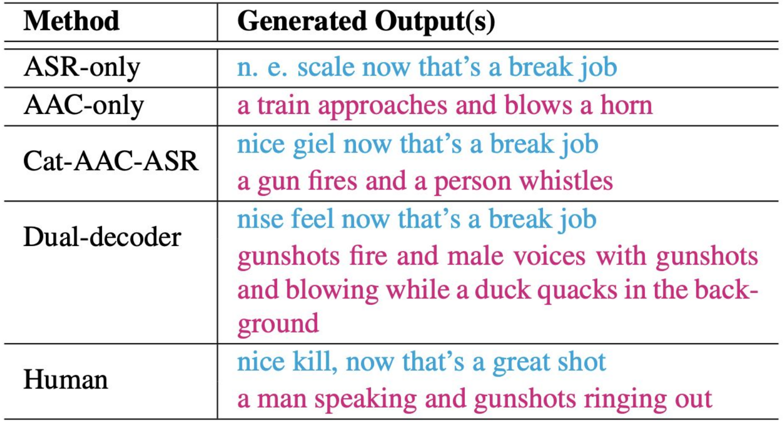
Performance for Each Mixing Weight



Method (without CTC)	**SPIDEr of dev-split**					**SPIDEr of eval-split**				
	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
AAC-only	0.309	0.322	0.292	0.283	0.279	0.292	0.280	0.293	0.291	0.252
Cat-ASR-AAC	0.328	0.311	0.343	0.348	0.354	0.297	0.309	0.337	0.339	0.326
Cat-AAC-ASR	**0.392**	**0.397**	**0.406**	**0.423**	**0.440**	**0.337**	**0.347**	**0.371**	**0.374**	**0.382**
Dual-decoder	0.268	0.289	0.320	0.327	0.327	0.229	0.235	0.243	0.242	0.239
Method (without CTC)	**CER of dev-split**					**CER of eval-split**				
	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
ASR-only	**9.0**	10.5	14.1	**17.8**	**22.2**	7.1	8.0	10.8	13.8	**17.4**
Cat-ASR-AAC	9.7	11.4	15.3	19.6	23.8	7.3	8.6	11.9	15.3	18.7
Cat-AAC-ASR	9.4	**10.4**	**13.9**	18.3	22.8	**6.4**	**7.2**	**9.9**	**13.4**	17.5
Dual-decoder	9.1	10.7	14.3	18.5	22.5	7.0	7.9	11.3	14.4	17.8
Method (with CTC)	**CER of dev-split**					**CER of eval-split**				
	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
ASR-only	5.7	6.5	9.3	12.6	**16.0**	**4.2**	5.0	6.9	**9.4**	**12.3**
Dual-decoder	**5.5**	**6.4**	**9.2**	**12.4**	16.3	**4.2**	**4.7**	**6.7**	9.6	12.4

Table 2. Comparison of models are trained without CTC (top, middle) and with CTC (bottom) and evaluated using CER (lower is better) and SPIDEr (higher is better) over the dev and eval splits for various mixing weights $\gamma \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$.

Testing with Real-world Noisy Speech



Method	Generated Output(s)
ASR-only	n. e. scale now that's a break job
AAC-only	a train approaches and blows a horn
Cat-AAC-ASR	nice giel now that's a break job a gun fires and a person whistles
Dual-decoder	nise feel now that's a break job gunshots fire and male voices with gunshots and blowing while a duck quacks in the background
Human	nice kill, now that's a great shot a man speaking and gunshots ringing out

Table 3. Comparison of transcripts (blue) and captions (pink) between human annotations and generations from independently and jointly trained models for a speech sample recorded in real-world.

Conclusion

- ASR and AAC are interdependent and both tasks generate coherent word sequences
- Proposed a process of synthesizing a dataset with both speech transcript and audio caption labels
- Also proposed joint modeling approaches that can outperform independently trained models
- Checkout: https://chintu619.github.io/Joint-ASR-AAC/ for synthetic dataset creation
- Now looking into annotating our own dataset with both transcription and captioning labels