JOINT SPEECH RECOGNITION AND AUDIO CAPTIONING

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

* Carnegie Mellon University, USA,
† Sony Group Corporation, Japan

ABSTRACT

Speech samples recorded in both indoor and outdoor environments are often contaminated with secondary audio sources. Most end-to-end monaural speech recognition systems either remove these background sounds using speech enhancement or train noise-robust models. For better model interpretability and holistic understanding, we aim to bring together the growing field of automated audio captioning (AAC) and the thoroughly studied automatic speech recognition (ASR). The goal of AAC is to generate natural language descriptions of contents in audio samples. We propose several approaches for end-to-end joint modeling of ASR and AAC tasks and demonstrate their advantages over traditional approaches, which model these tasks independently. A major hurdle in evaluating our proposed approach is the lack of labeled audio datasets with both speech transcriptions and audio captions. Therefore we also create a multi-task dataset by mixing the clean speech Wall Street Journal corpus with multiple levels of background noises chosen from the AudioCaps dataset. We also perform extensive experimental evaluation and show improvements of our proposed methods as compared to existing state-of-the-art ASR and AAC methods.

Index Terms— ASR, AAC, speech recognition, audio captioning, joint modeling

1. INTRODUCTION

Automatic speech recognition (ASR) is a well known task which decodes human speech as textual representations and has been a prominent research area for the past several decades [1]. End-to-end (E2E) ASR is a sequence-to-sequence task where a stream of acoustic features are converted into a sequence of words. These models typically use an encoder-decoder framework with connectionist temporal classification (CTC) loss [2, 3] or use RNN-Transducers [4, 5]. The introduction of attention mechanisms enabled the decoder to focus on relevant parts of the audio signal when generating long word sequences [6, 7]. Transformer layers with self-attention in both the encoder and decoder, have shown significant improvement in training speeds and modeling long-range dependencies [8].

Speech utterances captured in the real-world coexist with a wide variety of acoustic sources and are seldom pure. However most ASR systems focus only on the speech component of an audio signal and consider the non-speech sources as components to be disregarded during model learning [9, 10]. Synchronously modeling both speech and non-speech sources in a unified manner resembles the natural perception of the human auditory system. Such a unified model using Transformers was proposed by [11] to solve ASR, acoustic event detection (AED) and audio tagging (AT) tasks. Although tasks like AED and AT extract information of all constituent audio sources, the outputted sequence of events and tags lacks proper structure for straightforward human understanding [12].

Automated audio captioning (AAC) aims to provide the information of constituent audio sources and events in a structured and easily comprehensible manner i.e., a natural language description of a given audio waveform [12, 13]. Recently, Transformer based encoder-decoder frameworks are being employed to model the temporal structure of audio events [14, 15] AAC is an emerging research area with several applications, such as enriching the raw textual information provided by ASR during television broadcasting and video streaming. Such integration of ASR and AAC tasks can potentially improve the viewing experience of the hearing impaired.

Performance of an ASR model in a noisy environment depends on its ability to adapt to various background sounds and robustly infer the speech components. This process makes the ASR and AAC tasks interdependent and motivates us to treat them in a unified manner. Also, given the similarity of these tasks in transforming input acoustic signals into output word sequences, we formulate them as a multi-task problem. Our main contributions are as follows:

1. We present the very first attempt to jointly model ASR and AAC tasks, and propose various E2E Transformer frameworks to solve this multi-task problem.
2. Due to the lack of an audio dataset containing both transcript and caption labels, we prepare a synthetic multi-task dataset by combining clean speech samples and captioned non-speech samples.
3. We carefully evaluate the proposed jointly trained models on varying levels of background sounds and compare them with the independently trained ASR and AAC models.

2. INDEPENDENT MODELING

2.1. Automatic Speech Recognition

Popular E2E ASR systems are based on an encoder-decoder architecture, where the input acoustic features are decoded as speech transcripts. Let $X_{\text{spec}} \in \mathbb{R}^{F \times T}$ be an input spectrogram with frequency and time dimensions of F and T respectively, and W_{ASR} be the output word sequence. A neural network model with parameters Θ_{ASR}, aims to model the posterior distribution $P(W_{\text{ASR}}|X_{\text{spec}}, \Theta_{\text{ASR}})$, as shown in Fig. 1(a). Some state-of-the-art models follow a Transformer encoder-decoder framework, optimized using both CTC loss and attention loss [16]. Typically, pretrained language models (LM) are integrated with the decoder during inference to form coherent word sequences [17, 18]. However, as the purpose of this work is to compare the fundamental capability of independent and joint modeling approaches, we will not use LMs during inference.

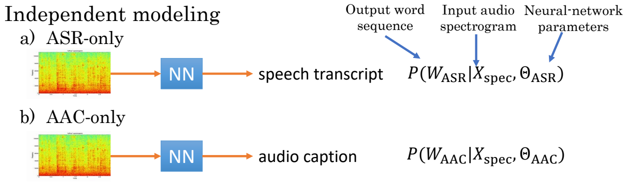

Fig. 1. Typical independent modeling of E2E ASR and AAC tasks

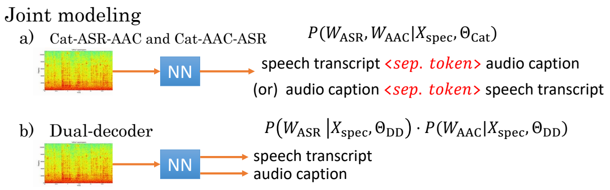
Joint modeling
a) Cat-ASR-AAC and Cat-AAC-ASR
$P(W_{ASR}, W_{AAC}|X_{spec}, \Theta_{Cat})$
NN
speech transcript <sep. token> audio caption
(or) audio caption <sep. token> speech transcript
b) Dual-decoder
$P(W_{ASR}|X_{spec}, \Theta_{DD}) \cdot P(W_{AAC}|X_{spec}, \Theta_{DD})$
NN
speech transcript
audio caption

Fig. 2. Joint modeling approaches of E2E ASR and AAC tasks

2.2. Automated Audio Captioning

In contrast to ASR systems, which primarily focus on speech contents, AAC systems must model all the acoustic sources in an input signal and output an intelligible textual description (caption) [12]. Neural network models for the AAC task aim to model the posterior distribution $P(W_{\text{AAC}}|X_{\text{spec}}, \Theta_{\text{AAC}})$, where W_{AAC} denotes the output word sequence in an audio caption, as shown in Fig. 1(b). As both ASR and AAC systems output word sequences, state-of-the-art AAC models also follow a Transformer based encoder-decoder framework, optimized using attention loss [15, 14]. Note that the CTC loss or RNN-Transducers are not particularly applicable to the AAC task, because the token sequence in a caption need not be temporally aligned with the input spectrogram frames. In this work, the transcripts and captions are tokenized as a sequence of characters.

3. MULTI-TASK DATASET SYNTHESIS

The inter-dependency of ASR and AAC tasks, in addition to their similar modeling techniques, motivates us to explore models that can be trained jointly on these two tasks. However an important challenge in realizing this joint modeling is the dearth of labeled real-world recordings with both a speech transcript and an audio caption. We overcome this hurdle by synthetically mixing clean speech samples from the Wall Street Journal (WSJ) corpus [19] with non-speech samples from AudioCaps dataset [12]. To support reproducible research, we will make our data generation scripts publicly available[1]. Our paper serves as a proof-of-concept to validate joint modeling of ASR and AAC tasks on a synthetic dataset, before creating a real-world multi-label dataset with transcriptions and audio captions.

WSJ corpus is a collective of the WSJ0 and WSJ1 datasets, totalling more than 37k clean speech samples and their transcripts. We choose the train_si284/dev_93/eval_92 splits of WSJ corpus as our train/dev/eval splits. The AudioCaps dataset contains 46k audio samples paired with human-annotated captions, and is a subset of AudioSet (a large-scale collection of 10-second audio clips from YouTube) [20]. Our goal is to prepare a dataset where an audio sample contains speech from a primary speaker in the presence of background noise. Keeping with this goal, we identify and remove all the speech samples present in the AudioCaps dataset. This is done by filtering audio samples whose captions contain the sub-strings 'speak' and 'talk' (identifying words such as speaks, talking, etc.). The filtering process discarded approximately 20k speech samples and retained 26k non-speech audio samples.

We created a synthetic multi-task dataset by randomly mixing WSJ samples with non-speech AudioCaps samples. Given a speech signal x_1 and a non-speech signal x_2, we perform mixing as:

$$x_{\text{mix}} = \hat{x}_1 + \gamma \cdot \hat{x}_2, \tag{1}$$

where $\hat{(.)}$ represents an amplitude normalization to $[-1, 1]$ range and γ denotes a scalar mixing weight. The duration of audio samples from WSJ can be up to 25 seconds, while those from AudioCaps are roughly 10 seconds. When $\hat{x}_1$ is longer than $\hat{x}_2$, we account for this duration mismatch during the mixing process in Eq. (1) by randomly choosing a segment of $\hat{x}_1$, such that the duration of chosen segment is equal to the duration of $\hat{x}_2$.

The transcript W_{ASR} of signal x_1 and caption W_{AAC} of signal x_2 are stored as labels for the normalized mixture signal $\hat{x}_{\text{mix}}$. Although the addition of a clean speech event to the sounds present in x_2 modifies the textual description of $\hat{x}_{\text{mix}}$, we overlook this issue from a practical standpoint. For instance, if W_{AAC} were to be '*a dog is barking*', then a human annotated caption for $\hat{x}_{\text{mix}}$ would probably be of the form '*a person is speaking while a dog is barking*'. Considering that all the mixture signals contain speech, their corresponding captions should ideally contain a phrase equivalent to '*a person is speaking*'. This inconsistency is common across all the mixture samples and can therefore be overlooked for our synthetic multi-task mixture dataset.

4. PROPOSED JOINT MODELING

Given a set of noisy speech signals contaminated with various background sounds, we aim to train a single model capable of generating both the transcript and caption labels. Inferring two output sequences from such a jointly trained model resembles an E2E multi-speaker ASR task [21, 22], without needing to accommodate for the permutation invariance among multiple speakers.

4.1. Concatenating Output Sequences

The independent ASR and AAC modeling frameworks discussed in Section 2 inherently follow a similar structure: acoustic feature encoding and word sequence decoding. Exploiting this similarity and the serialized output training proposed in [22], we propose an intuitive approach to our multi-task modeling: concatenating the word sequences W_{ASR} and W_{AAC} as follows:

$$W_{\text{Cat}} = \text{Concat}(\ W_{\text{ASR}}, \langle \textit{sep. token} \rangle\ , W_{\text{AAC}}\ ) \tag{2}$$

$$W_{\text{Rev}} = \text{Concat}(\ W_{\text{AAC}}, \langle \textit{sep. token} \rangle\ , W_{\text{ASR}}\ ), \tag{3}$$

where W_{Cat} and W_{Rev} denote the concatenated word sequences, and $\langle$*sep. token*$\rangle$ denotes a separation token. Eq. (2) signifies the scenario where W_{ASR} precedes W_{AAC} in the concatenated sequence, while Eq. (3) signifies a sequence concatenated in the reverse order. For definitive detection of transition from W_{ASR} to W_{AAC} or vice-versa during model inference, we incorporate a special separation token $\langle$*sep. token*$\rangle$ during concatenation as shown in Fig. 2(a).

Given the input X_{spec} and model parameters $\Theta_{\text{Cat}}, \Theta_{\text{Rev}}$ for $W_{\text{Cat}}, W_{\text{Rev}}$ respectively, we formulate the posterior distribution as:

$$P(W_{\text{Cat}}|X_{\text{spec}}, \Theta_{\text{Cat}}) = P(W_{\text{ASR}}|X_{\text{spec}}, \Theta_{\text{Cat}}) \cdot \\ P(W_{\text{AAC}}|W_{\text{ASR}}, X_{\text{spec}}, \Theta_{\text{Cat}}), \tag{4}$$

[1] https://chintu619.github.io/Joint-ASR-AAC/

$$P(W_{\mathrm{Rev}}|X_{\mathrm{spec}},\Theta_{\mathrm{Rev}}) = P(W_{\mathrm{AAC}}|X_{\mathrm{spec}},\Theta_{\mathrm{Rev}})\cdot \\ P(W_{\mathrm{ASR}}|W_{\mathrm{AAC}},X_{\mathrm{spec}},\Theta_{\mathrm{Rev}}). \quad (5)$$

We note that the formulations of Eq. (4) and Eq. (5) may appear as equivalent, but the evolving hidden state(s) of the neural network's recurrent architecture increases their dissimilarity with each decoded token. Henceforth, we denote the frameworks modeling W_{Cat} and W_{Rev} as 'Cat-ASR-AAC' and 'Cat-AAC-ASR' respectively.

Although the proposed sequence concatenation approach aims to model the true joint distribution of W_{ASR} and W_{AAC}, it suffers from several drawbacks. The decoding process is likely to output sub-optimal solutions due to a significant increase in the length of output sequence. Also, the constraint of generating a transcript and caption as part of the same output sequence removes the temporal alignment between W_{Cat} or W_{Rev} and $\hat{x}_{\mathrm{mix}}$. In-turn we lose the ability of optimizing our model using the CTC loss. Furthermore, the shared decoder is constrained to learn the unified token-transition probabilities within transcripts and captions, even though their individual token-transition probabilities need not be correlated.

4.2. Dual Output Decoding

Inspired by multi-task modeling in [11] and to alleviate the aforementioned drawbacks, we propose a shared encoder and dual-decoder framework to extract effective audio representations while separately decoding W_{ASR} and W_{AAC}. This approach assumes conditional independence between the transcript and caption given the input X_{spec} and dual-decoder model parameters Θ_{DD} when modeling the joint probability distribution. Specifically, we have:

$$P(W_{\mathrm{ASR}},W_{\mathrm{AAC}}|X_{\mathrm{spec}},\Theta_{\mathrm{DD}}) = \\ P(W_{\mathrm{ASR}}|X_{\mathrm{spec}},\Theta_{\mathrm{DD}})\cdot P(W_{\mathrm{AAC}}|X_{\mathrm{spec}},\Theta_{\mathrm{DD}}). \quad (6)$$

The flexibility of using two decoders allows the proposed model to individually learn the token transition distributions within transcripts and captions. The exponentially expanded search space of concatenated output model is now restored to output sequences of regular length. Finally, our proposed model also allows the use of the CTC loss when optimizing the ASR decoder as follows:

$$\mathcal{L} = \lambda\cdot\mathcal{L}_{\mathrm{ctc}} + (1-\lambda)\cdot\mathcal{L}_{\mathrm{att}}, \quad (7)$$

where $\mathcal{L}$, $\mathcal{L}_{\mathrm{ctc}}$ and $\mathcal{L}_{\mathrm{att}}$ denote the overall, CTC and attention losses respectively, and λ denotes the interpolation factor [23].

5. EXPERIMENTS

5.1. Dataset Preparation

Following the synthetic multi-task dataset preparation procedure detailed in Section 3, we randomly mix the 37k speech samples from WSJ with 26k non-speech AudioCaps samples using mixing weight γ. To mix the 836 total samples in development and evaluation splits of WSJ, we proportionately select 600 samples at random from AudioCaps that are not seen during training. We choose 5 discrete values for $\gamma \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$ to perform the above mixing process, aiming to evaluate our proposed methods at different levels of background sounds. This process results in 37k × 5 $\approx$ 187k training samples and roughly 4k dev and eval samples.

5.2. Model Training and Evaluation

We trained the independent and proposed joint modeling frameworks using the ESPNet toolkit [16]. The models compared during our experiments are as follows:

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.

1. ASR-only: trained to only output W_{ASR}
2. AAC-only: trained to only output W_{AAC}
3. Cat-ASR-AAC: outputs the word sequence W_{Cat} from Eq. (2)
4. Cat-AAC-ASR: outputs the word sequence W_{Cat} from Eq. (3)
5. Dual-decoder: trained to output both W_{ASR} and W_{AAC}

All the models discussed in this work follow the same network architecture to facilitate consistency during evaluation. Our Transformer encoder has 12 layers with 2048 units each, and 4 attention heads of 256 dimension. On the other hand, the Transformer decoder has 6 layers with 2048 units each. The proposed dual-decoder model in Section 4.2 employs two decoders with the above specifications. Our batch-size for training is set to 64, dropout to 0.1 and label smoothing weight to 0.1. We use the Noam optimizer to train for 100 epochs and perform model parameter averaging over epochs with the best 10 validation scores.

Since CTC is not applicable to the concatenated output model, as discussed in Section 4.1, we only perform experiments with the CTC loss for ASR-only and Dual-decoder models. In this set of experiments, we set the interpolation factor λ as 0.3 in Eq. (7). And during inference, we use beam-search with width 6 and CTC decoding weight of 0.3. When training without CTC, we observed that inference with a beam width of 1 gave the best results.

All the trained models are evaluated over the dev and eval mixture samples for all the 5 levels of background sounds. The generated speech transcripts are evaluated using traditional character and word error rates i.e., CER and WER. The captions are evaluated using captioning metrics: CIDEr [24], SPICE [25] and SPIDEr [26], where SPIDEr computes a simple average of CIDEr and SPICE scores. The n-gram cosine similarity between generated and target captions is computed by CIDEr, while their semantic similarity after lemmatization is computed by SPICE. Therefore, SPIDEr score balances the semantic and syntactic similarity during audio captioning.

6. RESULTS

This section evaluates the performance of models trained independently for ASR, AAC tasks and compares them with our proposed jointly trained models.

6.1. Independent vs. Joint Modeling

The overall scores of speech and captioning metrics for all mixture weights γ in Eq. (1), when training without the CTC is detailed in Table 1. Top and middle sections of Table 2 summarizes the performance comparison when training without CTC and evaluating over all 5 mixing weights. Expected trends include the increase of SPIDEr and CER scores with increasing mixture weight γ of non-speech audio signal. We also observe that CER for Cat-AAC-ASR model is better as compared to Cat-ASR-AAC. The latter infers a speech transcript after generating the caption, thereby demonstrating the ability

Method	**SPIDEr of dev-split**					**SPIDEr of eval-split**				
(without CTC)	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
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	**CER of dev-split**					**CER of eval-split**				
(without CTC)	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
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	**CER of dev-split**					**CER of eval-split**				
(with CTC)	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$	$\gamma = 0.8$
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 $\gamma \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$.

of Cat-AAC-ASR model to understand and utilize background audio information during ASR. Bottom section of Table 2 shows significant improvements when using CTC for speech transcription. It also empirically demonstrates the advantage of our joint modeling framework towards better speech recognition capability. For comparison, an ASR system trained independently on the clean WSJ corpus ($\gamma = 0$) achieves CER of 7.1/5.3 on the dev/eval sets without CTC and CER of 4.1/2.7 on dev/eval sets with CTC. Similarly, an AAC system trained independently on the clean AudioCaps dataset ($\gamma \to \infty$) achieves SPIDEr scores of 0.305/0.287 on dev/eval splits.

6.2. Testing with Real-world Speech Recordings

We acknowledge that it would be premature to conclusively state the superiority of our multi-task modeling based solely on improvements seen with synthetic mixtures. Therefore we additionally compare the model outputs over a few speech samples recorded in the presence of background sounds. These recordings were chosen from the speech samples present in AudioCaps dataset, which were filtered-out during mixing process.

For an audio sample captured during a duck hunt gameplay[2], the transcripts and captions generated from models trained without the CTC loss are compared with human annotations in Table 3. The 10 second audio sample actually starts with some gun shots, followed by a person speaking and more gun shots, while ducks quack in the background. Although the human annotated caption is short, our dual-decoder model is able to capture all the constituent audio sources. On the other hand, all the ASR outputs seem mostly similar, with our proposed Cat-AAC-ASR and dual-decoder models slightly outperforming the ASR-only model.

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.

7. CONCLUSIONS

ASR and AAC tasks generate speech transcripts and descriptive audio captions respectively, but share a commonality of outputting coherent word sequences from a given audio signal. As existing state-of-the-art models proposed for these tasks follow a similar Transformer based encoder-decoder framework, we explore the first known attempt at modeling ASR and AAC as a multi-task problem. Our proposed method takes advantage of the common Transformer frameworks and propose several joint modeling approaches. As there are no currently known audio datasets which are labeled with both speech transcripts and audio captions, we prepare a multi-task synthetic dataset by mixing the clean speech samples with non-speech captioned audio samples. Our extensive experiments over noisy speech samples with varying levels of background sounds, demonstrate the capability of our proposed multi-task models to match or even surpass the performance of independently trained ASR and AAC models.

8. ACKNOWLEDGEMENT

This work was supported in part by Sony Group Corporation and used the Extreme Science and Engineering Discovery Environment (XSEDE) [27], which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges system [28], which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC).

[2] https://youtu.be/8hSarhQXJbg?start=30&end=40

9. REFERENCES

[1] Dong Yu and Li Deng, *Automatic Speech Recognition.*, Springer, 2016.

[2] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*. PMLR, 2016, pp. 173–182.

[4] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[5] Jinyu Li, Rui Zhao, Zhong Meng, Yanqing Liu, Wenning Wei, Sarangarajan Parthasarathy, Vadim Mazalov, Zhenghao Wang, Lei He, Sheng Zhao, et al., "Developing RNN-T models surpassing high-performance hybrid models with customization capability," *arXiv preprint arXiv:2007.15188*, 2020.

[6] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[7] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. of ICASSP*. IEEE, 2016, pp. 4960–4964.

[8] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on Transformer vs RNN in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[9] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[10] Reinhold Haeb-Umbach, Shinji Watanabe, Tomohiro Nakatani, Michiel Bacchiani, Bjorn Hoffmeister, Michael L Seltzer, Heiga Zen, and Mehrez Souden, "Speech processing for digital home assistants: Combining signal processing with deep-learning techniques," *IEEE Signal processing magazine*, vol. 36, no. 6, pp. 111–124, 2019.

[11] Niko Moritz, Gordon Wichern, Takaaki Hori, and Jonathan Le Roux, "All-in-one transformer: Unifying speech recognition, audio tagging, and event detection.," in *INTERSPEECH*, 2020, pp. 3112–3116.

[12] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. of NAACL-HLT*, 2019, pp. 119–132.

[13] Konstantinos Drossos, Sharath Adavanne, and Tuomas Virtanen, "Automated audio captioning with recurrent neural networks," in *Proc. of WASPAA*. IEEE, 2017.

[14] Chaitanya Narisetty, Tomoki Hayashi, Ryunosuke Ishizaki, Shinji Watanabe, and Kazuya Takeda, "Leveraging state-of-the-art ASR techniques to audio captioning," Tech. Rep., DCASE2021 Challenge, July 2021.

[15] Xinhao Mei, Xubo Liu, Qiushi Huang, Mark D Plumbley, and Wenwu Wang, "Audio captioning transformer," *arXiv preprint arXiv:2107.09817*, 2021.

[16] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[17] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," *arXiv preprint arXiv:1706.02737*, 2017.

[18] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhijeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *Proc. of ICASSP*. IEEE, 2018, pp. 1–5828.

[19] Douglas B Paul and Janet Baker, "The design for the wall street journal-based CSR corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[20] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. of ICASSP*. IEEE, 2017, pp. 776–780.

[21] Xuankai Chang, Yanmin Qian, Kai Yu, and Shinji Watanabe, "End-to-end monaural multi-speaker ASR system without pretraining," in *Proc. of ICASSP*. IEEE, 2019, pp. 6256–6260.

[22] Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," *arXiv preprint arXiv:2003.12687*, 2020.

[23] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, 2017.

[24] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[25] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould, "Spice: Semantic propositional image caption evaluation," in *European conference on computer vision*. Springer, 2016, pp. 382–398.

[26] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy, "Improved image captioning via policy gradient optimization of spider," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 873–881.

[27] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "Xsede: Accelerating scientific discovery," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 62–74, Sept.-Oct. 2014.

[28] Nicholas A Nystrom, Michael J Levine, Ralph Z Roskies, and J Ray Scott, "Bridges: a uniquely flexible HPC resource for new communities and data analytics," in *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, 2015, pp. 1–8.