

TOWARDS WORD-LEVEL END-TO-END NEURAL SPEAKER DIARIZATION WITH AUXILIARY NETWORK

Yiling Huang, Weiran Wang, Guanlong Zhao, Hank Liao, Wei Xia, Quan Wang

Google LLC, USA

{ yilinghuang, weiranwang, guanlongzhao, hankliao, ericwxia, quanw }@google.com

ABSTRACT

While standard speaker diarization attempts to answer the question “who spoken when”, most of relevant applications in reality are more interested in determining “who spoken what”. Whether it is the conventional modularized approach or the more recent end-to-end neural diarization (EEND), an additional automatic speech recognition (ASR) model and an orchestration algorithm are required to associate the speaker labels with recognized words. In this paper, we propose Word-level End-to-End Neural Diarization (WEEND) with auxiliary network, a multi-task learning algorithm that performs end-to-end ASR and speaker diarization in the same neural architecture. That is, while speech is being recognized, speaker labels are predicted simultaneously for each recognized word. Experimental results demonstrate that WEEND outperforms the turn-based diarization baseline system on all 2-speaker short-form scenarios and has the capability to generalize to audio lengths of 5 minutes. Although 3+speaker conversations are harder, we find that with enough in-domain training data, WEEND has the potential to deliver high quality diarized text.

Index Terms— Speaker diarization, ASR, word-level, end-to-end, auxiliary network

1. INTRODUCTION

Speaker diarization is the task of partitioning speech into homogeneous segments according to speaker identities. The traditional approach is a combination of multiple individually trained modules, including voice activity detection (VAD), speaker turn segmentation, speaker encoder and clustering. Each module has been extensively studied to improve speaker diarization, including personalized VAD [1], better speaker turn detection [2], fine-tuning speaker encoders for specific scenarios (e.g. ECAPA-TDNN [3] for short queries), and various clustering algorithms [4, 5, 6]. More recently, the research community has been exploring supervised end-to-end approaches including UIS-RNN [7], DNC [8], frame-level end-to-end neural diarization (EEND) [9], and its other variants [10, 11, 12, 13, 14]. Other methods are described and discussed in literature reviews and tutorials [15, 16].

Most of the above mentioned speaker diarization systems output timestamped segment-level speaker labels (i.e. “who spoke when”), which are usually not useful by themselves. For most real-world applications, these speaker labels need to be associated with words recognized by an ASR system (i.e. “who spoke what”). This involves a complicated multi-module architecture with an orchestration algorithm to merge ASR and diarization results based on segment timestamps. Both modules are also required to be synchronized to have similar latency for the best results. To address these challenges, there

have been a few pioneering proposals for joint modeling of word-level speaker diarization with ASR, summarized in Section 2.

Inspired by the multi-output, multi-task learning work [17] and many other ASR-auxiliary joint learning studies [18, 19], we extend the ASR-auxiliary joint modeling architecture to include an auxiliary network for speaker diarization, with a separate encoder and joint network for predicting speaker labels. We assess our method on various test scenarios: public and simulated data, short-form and long-form audios, 2-speaker and 3+ speaker scenarios, etc. Experiments show that WEEND significantly outperforms the turn-based diarization baseline on Callhome by 25%, demonstrates superior quality across all 2-speaker short-form test scenarios and generalizes up to 5 minutes of 2-speaker long-form audios with no performance degradation. For speech that involve 3+speakers, WEEND is still capable of predicting speaker labels well provided that it is trained on sufficient in-domain data.

2. RELATED WORK

Shafey et al. [20] proposed inserting speaker role tags into the transcripts and training like ASR. However, the problem was constrained to 2-speaker doctor-patient conversations. It is solving a word-level doctor-patient classification problem instead of a generic speaker diarization problem. Another related category of work is speaker attributed ASR (SA-ASR), which typically takes the additional input of speaker profiles and identifies speaker profile indices based on an attention mechanism [21, 22, 23, 24]. In the absence of enrolled speaker profiles, the SA-ASR model performs speaker clustering on internal embeddings [25]. Moreover, SA-ASR involves an inherent turn detection where it segments speech according to speaker change points. In addition, target speaker ASR (TS-ASR) [26, 27, 28, 29] can also be considered as diarizing target speaker speech via enrolled speaker embedding extraction.

The main contributions and novelty of our work lie in the following aspects: (1) We propose Recurrent Neural Network Transducer (RNN-T) based ASR-diarization multi-task learning, where both tasks are strongly coupled by sharing blank logits. (2) Our approach leads to a much simpler pipeline, with no requirement of speaker profiles, enrollment or clustering. (3) We can make use of pre-trained ASR systems to quickly adapt to the diarization task, without affecting ASR performance.

3. SYSTEM DESCRIPTIONS

As discussed in Section 1, diarization is particularly helpful when associated with recognized words. Speaker labels are also natively coupled with words. Thus, we model ASR and diarization together. For each utterance, we aim at recognizing the speech words and pre-

dicting the corresponding speakers simultaneously. For the target sequence, we tokenize the transcript with a wordpiece model and meanwhile construct a same-length speaker label sequence. Within each utterance, we map raw speaker labels (e.g. “speaker:A”) to integer-indexed speaker labels in a “first come, first serve”, order-based fashion. That is, the N th speaker that starts speaking is labeled as $\langle \text{spk} : N \rangle$. In the following sections, we describe the blank sharing multi-output setup in Section 3.1. Section 3.2 introduces our proposed method and its main modifications.

3.1. RNN-T with multi-output joint networks

We follow the RNN-T ASR architecture in Wang et al. [17]. Specifically, the joint network of an RNN-T model [30] fuses audio features extracted by the encoder with the text features extracted by the prediction network. Formally, let the encoder output be $[f_0, \dots, f_{T-1}]$ and the prediction network output be $[g_0, \dots, g_{U-1}]$, where $f_t \in \mathcal{R}^{D_a}$, $g_u \in \mathcal{R}^{D_l}$. t and u denote the time and label sequence indices, D_a and D_l denote acoustic and text feature dimensions. The ASR symbol space \mathcal{Y} consists of a special $\langle \text{blank} \rangle$ token for non-emission and a set of $V - 1$ non-blank wordpieces, i.e., $\mathcal{Y} = \{y^0 = \langle \text{blank} \rangle, y^1, \dots, y^{V-1}\}$. The joint network hidden embedding $h_{t,u}$ is merged from the acoustic and text features:

$$h_{t,u} = P \cdot f_t + Q \cdot g_u + b_h \in \mathcal{R}^{D_h} \quad (1)$$

where P, Q are projection matrices and b_h is the bias term. The raw logits $s_{t,u}$ before softmax are computed:

$$s_{t,u} = A \cdot \tanh(h_{t,u}) + b_s \in \mathcal{R}^V \quad (2)$$

where A is the projection matrix, b_s is the bias term. We use the hybrid auto-regressive transducer model [31] and the factorized posterior probability distribution over \mathcal{Y} can be formulated as:

$$P_{t,u}(y^v | f_{0:t}, g_{0:u}) = (1 - b_{t,u}) \cdot \text{softmax}(s_{t,u}[1:])[v - 1] \quad (3)$$

for $v = 1, \dots, V - 1$, with $y^v \neq \langle \text{blank} \rangle$, where the factorized blank distribution $b_{t,u}$ is defined as:

$$b_{t,u} := P_{t,u}(\langle \text{blank} \rangle | f_{0:t}, g_{0:u}) = \text{sigmoid}(s_{t,u}[0]) \quad (4)$$

To extend the RNN-T architecture for auxiliary tasks, we introduce additional last linear layer parameters A^{aux} and b_s^{aux} . Blank logits are shared between ASR and the auxiliary task to ensure output synchronization across tasks. Denote the auxiliary task output space is $\mathcal{Y}_{\text{aux}} = \{\langle \text{blank} \rangle, y_{\text{aux}}^1, \dots, y_{\text{aux}}^{V_{\text{aux}}-1}\}$, where V_{aux} is the size of the auxiliary label space including the shared blank. We re-use the blank logits from ASR (2) and the auxiliary task raw logits can be expressed this way:

$$s_{t,u}^{\text{aux}} = [s_{t,u}[0], A^{\text{aux}} \cdot \tanh(h_{t,u}) + b_s^{\text{aux}}] \in \mathcal{R}^{V_{\text{aux}}}. \quad (5)$$

For the inference procedure, ASR blank emissions are directly shared. Whenever the ASR beam search emits a non-blank, we apply softmax on the auxiliary logits $s_{t,u}^{\text{aux}}$ to obtain the probabilities over the auxiliary label space \mathcal{Y}_{aux} :

$$P_{t,u}(\mathcal{Y}_{\text{aux}} | \text{non-blank}, f_{0:t}, g_{0:u}) = \text{softmax}(s_{t,u}^{\text{aux}}[1:]) \quad (6)$$

and select the auxiliary label with $\text{argmax}(\cdot)$ on the softmax logits.

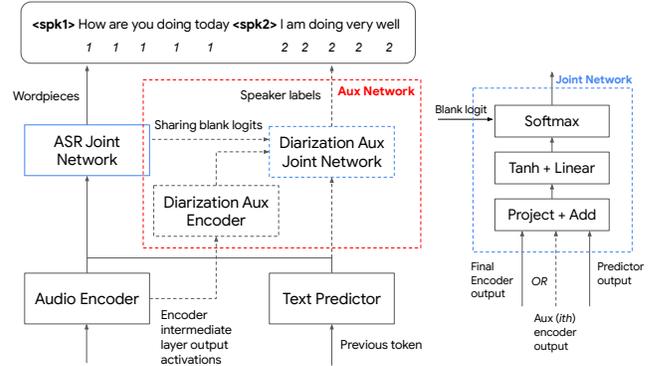


Fig. 1: Diagram of the proposed ASR-diarization joint architecture with an auxiliary network of an auxiliary encoder and a separate joint network sharing ASR blank logits. Dotted lines indicate the newly added designs. Solid outlined modules are frozen during training.

3.2. Proposed joint ASR and diarization system

In the context of speaker diarization, auxiliary labels are now speakers. The synchronization between ASR and diarization perfectly enables the word-level speaker diarization task. Therefore, we further extend the multi-output joint architecture from Section 3.1, and propose this neural architecture with the addition of an auxiliary network in Figure 1 including an auxiliary diarization encoder, intermediate layer activations wiring and an auxiliary joint network. The necessity of these is further discussed in Section 4.6.

3.2.1. Auxiliary diarization encoder

We insert a second, auxiliary encoder for diarization, which generates a new encoder feature output $f_t^{\text{aux}} \in \mathcal{R}^{D_{\text{aux}}}$ via Eq. (1). The motivation of this encoder is to extract speaker-related features. The encoder can be any model architecture capable of modeling temporal context, such as LSTM [32], transformer or conformer.

3.2.2. Intermediate activations

We feed the ASR encoder intermediate activations as the input to the auxiliary diarization encoder. According to these work [19, 33], we expect the model to perform the best when using intermediate layer outputs instead of raw audio features or last layer outputs.

3.2.3. Auxiliary joint network

We define a separate auxiliary joint network for diarization label sequence prediction. Following the notation (1), we introduce the auxiliary parameters $P_{\text{aux}}, Q_{\text{aux}}$ and $b_{h_{\text{aux}}}$ to learn the hidden embedding from the acoustic and text features, as well as the final projection parameters A^{aux} and b_s^{aux} .

4. EXPERIMENTS

4.1. Datasets

Our public datasets are listed in the first 3 rows of Table 1:

1. AMI [34]: we use the official “Full-corpus-ASR partition” training and test subsets. The speaker label ground truth was obtained based on the word-level annotations in the v1.6.2 AMI manual annotation release.

Table 1: Diarization public and simulated datasets statistics

Datasets	Domain	# Spk	Avg length (sec)		Total hours (hr)	
			Train	Eval	Train	Eval
AMI	Meeting	3-4	15/30/60	2039	81	9.1
Callhome	Telephone	2	15/30/60	301	14	1.7
Fisher	Telephone	2	15/30/60	600	1920	28.7
Sim 2spk	Read	2	57.9	36.1	6434	39.7
Sim 3spk	Read	3	68.5	43.1	7137	43.3
Sim 4spk	Read	4	94.2	59.4	9848	57.0

2. Callhome [35]: we use the Callhome American English Speech official training and evaluation subsets.
3. Fisher [36]: we withhold a test subset of 172 utterances¹ from Fisher English Training Speech and use the rest for training.

The training data are segmented into 15, 30 and 60 second segments for model training. The segmentation avoids chopping in the middle of a sentence. For the test splits, we evaluate not only the original full-length audios but also their short-form segments of 30, 60 and 120 seconds. The official metadata RTTMs are converted to target word and speaker label sequences.

Besides public data, we simulate multi-speaker utterances from LibriSpeech for data augmentation to mitigate the lack of training data. For each simulated conversation, we sample M unique speakers and N utterances from each speaker, and randomly drop $K = 0, 1, 2$ utterances for variety. Remaining samples are concatenated in random order, with inserted pause (uniform from 0.2 to 1.5 seconds) and cross-fade (uniform from 0 to 0.2 seconds). There are both real and fake speaker turns from this simulation. Simulated data statistics are also listed in Table 1. For training sets, we sample from LibriSpeech train-clean-100h, train-clean-360h and train-other-500h. For test sets, we sample from test-clean and test-other.

4.2. Model architecture

We extract 128-dimensional log-Mel spectrogram features using a 32ms Hamming window with a hop length of 10ms. Every 4 consecutive frames are stacked and sub-sampled by a factor of 3 to generate 512-dimensional features at 30ms frame rate. The input features are first fed to the causal ASR encoder, which consists of 12 conformer layers [37] of 512 dimensions, with 8-head attention, convolutional kernel size of 15 and a left context of 23 frames. Funnel pooling [38], with a downsampling factor of 2, is placed after the fourth conformer layer. The ASR encoder outputs $D_a = 512$ feature dimensions. The decoder is an embedding-based prediction network which computes language model features of $D_l = 640$ dimensions, based on two previous non-blank tokens [39]. The ASR joint network hidden dimension is $D_h = 640$. A last linear layer projects the hidden dimension to the wordpiece model vocabulary size $V = 4096$.

As for the auxiliary network, the 5th ASR Conformer layer outputs are fed to our LSTM encoder, which has a stacked of 9 layers, each layer containing 1024 hidden nodes and 512 output nodes. The auxiliary joint is set to have a hidden dimension of $D_h^{\text{aux}} = 640$, and the same prediction network outputs are used in the auxiliary joint. We pre-define a speaker set from 1 to $N = 8$, so the last linear layer projects the joint hidden embedding to $V_{\text{aux}} - 1 = 8$. For training, we initialize the ASR audio encoder, text predictor, and joint network from this pre-trained ASR model [39] and freeze the parameters of

¹https://github.com/google/speaker-id/blob/master/publications/Multi-stage/evaluation/Fisher/eval_whitelist.txt

Table 2: ASR and diarization performance of the baseline and proposed models. We report WERs (%) followed by the detailed breakdowns of substitution (S), deletion (D) and insertion (I) error rates.

Testsets	WER (S/D/I)	WDER (%)	
		Baseline	Proposed
Callhome	45.9 (12.8/9.7/23.3)	10.3	7.7
Fisher	20.5 (8.7/10.4/1.4)	3.6	8.0
AMI	29.6 (8.9/19.9/0.8)	8.7	50.0
Sim 2spk	8.1 (6.4/1.0/0.7)	4.2	4.1
Sim 3spk	8.3 (6.5/1.0/0.8)	4.2	3.6
Sim 4spk	8.1 (6.4/1.0/0.7)	4.5	5.1

these components (i.e. only the auxiliary network parameters are updated during training). The loss function is a standard RNN-T loss but we only include the speaker label RNN-T loss (with the shared blank logits), because the ASR RNN-T loss acts more like a scaling factor when ASR is frozen.

4.3. Baseline: turn-based diarization

We set up the turn-based diarization baseline following the “turn-to-diarize” system [2, 40] without pairwise speaker turn constraints and apply multi-stage clustering [41]. We pair it with the same RNN-T ASR model described in Section 4.2 to retrieve word-level speaker labels, specifically by assigning speakers to the recognized words based on the maximum speaker overlap in duration for each word.

4.4. Metrics

We report the Word Error Rate (WER) for ASR quality, and the Word Diarization Error Rate (WDER) from [20] for diarization quality. This WDER is a time-invariant diarization error metric that does not take time boundaries into consideration, and it suits the word-level end-to-end diarization problem better. In addition, datasets like AMI have lots of overlaps but the ASR system by itself does not support overlapping speech. This leads to considerable label confusion around overlapping speech and quick speaker changes. Hence, for AMI in this paper, we evaluate and report a modified WDER² which does not count words that overlap with any other word in the ground truth. We gather statistics of how many words are dropped to ensure there are still enough words left from each utterance. In our evaluation, the word drop percentage distribution on AMI has a mean of 10% and standard deviation of 14%.

4.5. Experimental results

The ASR and diarization quality is reported in Table 2. For ASR, Callhome has high insertion error because its ground truth transcript contains many missing words. The deletion error rate is high due to the fact that ASR does not support overlapping speech. The annotation standard of the diarization datasets (e.g. “mhm”, “y- ye- yes”) adds to the deletions and substitutions. On simulated datasets, the substitution rate is high because ASR is not trained on read speech like LibriSpeech. For diarization, our model outperforms the baseline on 5-min Callhome test data by 25%. The performance degrades on longer audios such as 10-min Fisher test utterances. AMI has the hardest test cases: longest audio, more speakers, overlapping speech. We investigate each of those aspects in-depth.

²The modified metric algorithm and unmodified AMI metrics are reported at <https://github.com/google/speaker-id/blob/master/publications/WEEND/README.md>

Table 3: Short-form test WDER (%) on various audio duration.

Testsets	Short-form Lengths (s)	WDER (%)	
		Baseline	Proposed
Callhome	30	13.6	9.3
	60	9.8	8.9
	120	10.5	8.9
Fisher	30	8.6	3.8
	60	4.8	3.7
	120	4.0	3.7
AMI	30	10.1	9.9
	60	6.7	13.3
	120	8.0	18.8

Table 4: Pre-segmented short-form AMI WDER (%) of the baseline and proposed models, breakdown by reference number of speakers. For each testset, we compute the WDER for each group of utterances with the same number of ground truth speakers. For the evaluation on 120-sec segments, since there are only 6 speaker test examples, we do not list these results.

AMI Lengths	Baseline WDER (%)				Proposed WDER (%)			
	1spk	2spk	3spk	4spk	1spk	2spk	3spk	4spk
30-sec	18.6	10.0	8.8	8.4	1.1	5.8	10.1	15.5
60-sec	10.8	6.3	5.6	6.9	0.8	5.2	12.2	17.1
120-sec	-	6.4	4.4	9.3	-	9.8	15.8	20.8

4.5.1. Short-form and audio duration

Table 3 presents the short-form performance. Our proposed model outperforms the baseline on short-form Callhome and Fisher significantly, particularly on 30s segments by 32% and 56%. However, the limitation is that the performance degrades if the audio is very long (e.g. Fisher), and it is most notable on AMI test data, which is the longest. This limitation stems from the mismatch between training segments and the full length test segments. In contrast, the baseline does not suffer from this issue because it is an unsupervised, clustering based model. In fact, the baseline quality is sub-optimal when the audio is only 30 seconds. This is because each segment can be 6 seconds, and a 30-second utterance might end up with only 5 segments, which is not enough for clustering to achieve good results.

4.5.2. Variable number of speakers

To understand the effect of number of speakers, we break down the composition of AMI WDER bucketed by the number of speakers. Table 4 shows that our model performs better than the baseline on 1 or 2 speaker cases (except for the 120-sec testset). The majority of errors come from 3 or 4 speaker cases. Since Table 2 proves that the model is indeed capable of learning 3+speaker diarization on simulated testsets, we believe the poor performance comes from (1) very limited amount of in-domain training data (only 80 hours) (2) quick speaker changes in meeting conversations. The second point matches our loss analysis observations: if there is no interruption or background speaker, speaker labels are predicted correctly, consistently. But speaker predictions have much higher chances to be wrong around quick speaker change points. This is likely because ASR in our proposed system only emits a single speaker speech on overlapping speech. It confuses model training with which speaker label to predict on nearby words.

Table 5: Impact of intermediate layer selection on WDER (%). Callhome is abbreviated as CH. Average numbers are reported for simulated and short-form AMI.

Intermediate Layer Selection	CH	Fisher	Sim	AMI Short
0th Conf layer (features)	23.8	24.5	10.4	22.8
5th Conf layer (proposed)	7.7	8.0	4.3	14.0
12th Conf layer (last)	33.6	37.3	46.9	27.5

Table 6: Training data augmentation impact on WDER (%). The second row excludes simulated data from training. The last row further drops 30/60s training segments, i.e. only trained on 15s data.

Model	CH	CH Short	Fisher	Fisher Short	Sim	AMI Short
Proposed	7.7	9.0	8.0	3.7	4.3	14.0
-Simulated	11.6	9.8	12.3	5.4	22.2	19.0
-30/60s segs	28.8	22.5	22.1	15.7	26.8	26.2

4.6. Ablation studies

We explore how much the network architecture affects model performance. If we completely remove the auxiliary encoder (i.e. use the ASR encoder output along with a separate joint network directly), the model simply does not learn diarization properly. If we keep the auxiliary encoder, intermediate layer output selection leads to different outcomes. As shown in Table 5, neither the first layer nor the last layer works well. This aligns with our expectation and conclusions from previous work [19, 33]. ASR encoder tends to discard speaker knowledge towards the last layer. Raw features are also hard for training to converge to the optimal space. There exists a sweet spot where a certain intermediate layer works the best.

Table 6 summarizes our data ablation studies. Even though the simulated data is from a different domain, it still effectively mitigates the public data insufficiency. Fisher and Callhome short-form improvements are the least, most likely due to the fact that we have enough 2-speaker telephony training data from Fisher (almost 2k hours). Furthermore, if the training data only include 15s segments, the model quality degrades dramatically on all testsets. This matches the audio duration generalization discussion in Section 4.5.1 and suggests that we should include longer training segments if feasible.

5. CONCLUSIONS

This paper proposed and studied word-level end-to-end neural diarization via auxiliary networks, without involving turn-based segmentation, speaker profiles or clustering. This approach presents promising opportunities over conventional unsupervised methods, where we observed superior performance on 2-speaker short-form scenarios. The limitation is that due to the lack of 3+speaker, long-form in-domain training data, the model does not generalize well enough to those cases. In the future, we would like to investigate chunk-aware learning for long-form training with historical context for long-form generalization. Additionally it is essential to develop advanced data augmentation techniques to simulate large amounts of in and out of domain conditions with arbitrary number of speakers. Lastly, this architecture can be further extended to be support overlapping speech with serialized output training (SOT).

6. REFERENCES

- [1] Ivan Medennikov et al., “Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario,” in *Interspeech*, 2020.
- [2] Wei Xia et al., “Turn-to-Diarize: Online speaker diarization constrained by transformer transducer speaker turn detection,” in *ICASSP*. IEEE, 2022, pp. 8077–8081.
- [3] Nauman Dawalatabad et al., “ECAPA-TDNN embeddings for speaker diarization,” *arXiv preprint arXiv:2104.01466*, 2021.
- [4] Quan Wang et al., “Speaker diarization with LSTM,” in *ICASSP*. IEEE, 2018, pp. 5239–5243.
- [5] Dimitrios Dimitriadis et al., “Developing on-line speaker diarization system,” in *Interspeech*, 2017, pp. 2739–2743.
- [6] Daniel Garcia-Romero et al., “Speaker diarization using deep neural network embeddings,” in *ICASSP*. IEEE, 2017, pp. 4930–4934.
- [7] Aonan Zhang et al., “Fully supervised speaker diarization,” in *ICASSP*. IEEE, 2019, pp. 6301–6305.
- [8] Qiuqia Li et al., “Discriminative neural clustering for speaker diarisation,” in *SLT*. IEEE, 2021.
- [9] Yusuke Fujita et al., “End-to-end neural speaker diarization with permutation-free objectives,” in *Interspeech*, 2019, pp. 4300–4304.
- [10] Shota Horiguchi et al., “End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors,” *arXiv preprint arXiv:2005.09921*, 2020.
- [11] Yusuke Fujita et al., “Neural speaker diarization with speaker-wise chain rule,” *arXiv preprint arXiv:2006.01796*, 2020.
- [12] Yawen Xue et al., “Online end-to-end neural diarization with speaker-tracing buffer,” in *SLT*. IEEE, 2021, pp. 841–848.
- [13] Eunjung Han et al., “BW-EDA-EEND: Streaming end-to-end neural speaker diarization for a variable number of speakers,” in *ICASSP*. IEEE, 2021, pp. 7193–7197.
- [14] Keisuke Kinoshita et al., “Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds,” in *ICASSP*. IEEE, 2021, pp. 7198–7202.
- [15] Tae Jin Park et al., “A review of speaker diarization: Recent advances with deep learning,” *Computer Speech & Language*, vol. 72, pp. 101317, 2022.
- [16] Chao Zhang et al., “Speaker diarization: A journey from unsupervised to supervised approaches,” *Odyssey: The Speaker and Language Recognition Workshop*, 2022, Tutorial session.
- [17] Weiran Wang et al., “Multi-output RNN-T joint networks for multi-task learning of ASR and auxiliary tasks,” in *ICASSP*. IEEE, 2023, pp. 1–5.
- [18] Chunxi Liu et al., “Improving RNN transducer based ASR with auxiliary tasks,” in *SLT*. IEEE, 2021, pp. 172–179.
- [19] Jicheng Zhang et al., “E2E-based multi-task learning approach to joint speech and accent recognition,” *arXiv preprint arXiv:2106.08211*, 2021.
- [20] Laurent El Shafey et al., “Joint speech recognition and speaker diarization via sequence transduction,” in *Interspeech*, 2019, pp. 396–400.
- [21] Naoyuki Kanda et al., “Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers,” *arXiv preprint arXiv:2006.10930*, 2020.
- [22] Naoyuki Kanda et al., “Minimum bayes risk training for end-to-end speaker-attributed ASR,” in *ICASSP*. IEEE, 2021, pp. 6503–6507.
- [23] Naoyuki Kanda et al., “End-to-end speaker-attributed ASR with transformer,” *arXiv preprint arXiv:2104.02128*, 2021.
- [24] Naoyuki Kanda et al., “Streaming speaker-attributed ASR with token-level speaker embeddings,” *arXiv preprint arXiv:2203.16685*, 2022.
- [25] Naoyuki Kanda et al., “Investigation of end-to-end speaker-attributed ASR for continuous multi-talker recordings,” in *SLT*. IEEE, 2021, pp. 809–816.
- [26] Katerina Zmolikova et al., “Speaker-aware neural network based beamformer for speaker extraction in speech mixtures,” in *Interspeech*, 2017, pp. 2655–2659.
- [27] Marc Delcroix et al., “Single channel target speaker extraction and recognition with speaker beam,” in *ICASSP*. IEEE, 2018, pp. 5554–5558.
- [28] Marc Delcroix et al., “End-to-end SpeakerBeam for single channel target speech recognition,” in *Interspeech*, 2019, pp. 451–455.
- [29] Naoyuki Kanda et al., “Auxiliary interference speaker loss for target-speaker speech recognition,” *arXiv preprint arXiv:1906.10876*, 2019.
- [30] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv:1211.3711*, 2012.
- [31] Ehsan Variani et al., “Hybrid autoregressive transducer (HAT),” in *ICASSP*, 2020.
- [32] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [33] Jicheng Zhang et al., “Intermediate-layer output regularization for attention-based speech recognition with shared decoder,” *arXiv preprint arXiv:2207.04177*, 2022.
- [34] Jean Carletta et al., “The AMI meeting corpus: A pre-announcement,” in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.
- [35] A Canavan et al., “CALLHOME American English speech LDC97S42,” LDC Catalog. Philadelphia: Linguistic Data Consortium, 1997.
- [36] Christopher Cieri et al., “The Fisher corpus: A resource for the next generations of speech-to-text,” in *LREC*, 2004, vol. 4, pp. 69–71.
- [37] Anmol Gulati et al., “Conformer: Convolution-augmented transformer for speech recognition,” in *Interspeech*, 2020.
- [38] Zihang Dai et al., “Funnel-transformer: Filtering out sequential redundancy for efficient language processing,” *Advances in Neural Information Processing Systems*, 2020.
- [39] R. Botros et al., “Tied & reduced RNN-T decoder,” in *Interspeech*, 2021.
- [40] Guanlong Zhao et al., “Augmenting transformer-transducer based speaker change detection with token-level training loss,” in *Proc. ICASSP*, 2023.
- [41] Quan Wang et al., “Highly efficient real-time streaming and fully on-device speaker diarization with multi-stage clustering,” *arXiv preprint arXiv:2210.13690*, 2022.