Interspeech 2024
1-5 September 2024, Kos, Greece

# On the Success and Limitations of Auxiliary Network Based Word-Level End-to-End Neural Speaker Diarization

*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 spoke when", many realistic applications are interested in determining "who spoke what". In both the conventional modularized approach and the more recent end-to-end neural diarization (EEND), an additional automatic speech recognition (ASR) model and an orchestration algorithm are required to associate speakers 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 architecture by sharing blank logits. Such a framework allows easily adding diarization capabilities to any existing RNN-T based ASR models without Word Error Rate (WER) regressions. Experimental results demonstrate that WEEND outperforms a strong turn-based diarization baseline system on all 2-speaker short-form scenarios, with the capability to generalize to audio lengths of 5 minutes.

**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 conventional 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 [4–6]. 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 [10–14]. Other methods are described and discussed in literature reviews and tutorials [15, 16].

Most of the speaker diarization systems above mentioned output timestamped segment-level speaker labels (i.e. "who spoke when"), which are usually less useful for applications such as meeting or recording summarization. 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 learning studies [18, 19], we extend the ASR-auxiliary multi-task architecture to include an auxiliary network for speaker diarization, with a separate encoder and joint network for predicting speaker labels. We freeze the pretrained ASR and assess our method on various test scenarios: public and simulated data, short-form and long-form audios, 2-speaker and 3+ speaker scenarios. 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 cases and generalizes up to 5 minutes of 2-speaker long-form audios with no performance degradation. For speech that involve 3+speakers, WEEND is capable of predicting speakers but requires sufficient in-domain data (as shown by the training and evaluation on large amounts of simulated data). End-to-end speaker diarization on audios that are much longer (e.g. over 30 minutes) entails cross-segment historical context in training and remains to be a challenge.

## 2. Related work

Shafey et al. [20] proposed directly inserting speaker role tags into the transcripts and training as a standard ASR. However, the problem was constrained to a 2-speaker doctor-patient classification problem, which does not work in generic speaker diarization tasks[1]. 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 [21–24]. 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. Target speaker ASR (TS-ASR) [26–29] can also be considered as diarizing target speaker speech via enrolled speaker embedding extraction. These methods that rely on extracting speaker embeddings utilize sensitive biometric information which can be exploited for unintended purposes and are sub-optimal from a privacy point of view [30].

The main contributions and novelty of our work lie in the following aspects: (1) We propose a novel Recurrent Neural Network Transducer (RNN-T) based ASR-diarization multi-task learning framework, where both tasks are strongly coupled by sharing blank logits. (2) Our approach adds diarization capabilities to any frozen RNN-T based ASR model with no WER regression. (3) Our paper presents a much simpler pipeline with less privacy concerns, by removing components for enrollment,

[1] `https://github.com/google/speaker-id/tree/master/publications/WEEND#baseline-with-inserted-speaker-tags`

10.21437/Interspeech.2024-561

speaker profiles, and cross-segment clustering.

## 3. System descriptions

For each utterance, we recognize the spoken words and predict 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 `<spk:N>`. Note that permutation invariant training (PIT) is also promising, but we will leave that for future work. 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 [31] 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 `<blank>` token for non-emission and a set of $V-1$ non-blank wordpieces, i.e., $\mathcal{Y} = \{y^0 = \texttt{<blank>}, y^1, \dots, y^{V-1}\}$. The joint network hidden embedding $h_{t,u}$ is merged from the acoustic and text features:

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

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

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

where $A$ is the projection matrix, $b_s$ is the bias term. We use the hybrid auto-regressive transducer model [32] and the factorized posterior probability distribution over $\mathcal{Y}$ can be formulated as:

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

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

$$b_{t,u} := P_{t,u}(\texttt{<blank>} | f_{0:t}, g_{0:u}) = \text{sigmoid}(s_{t,u}[0]) \tag{4}$$

To extend the RNN-T architecture for auxiliary tasks, we introduce additional last linear layer parameters $A^{\text{aux}}$ and $b_s^{\text{aux}}$. Blank logits are shared between ASR and the auxiliary task to ensure output synchronization across tasks. Denote the auxiliary task output space is $\mathcal{Y}_{\text{aux}} = \{\texttt{<blank>}, y^1_{\text{aux}}, \dots, y^{V_{\text{aux}}-1}_{\text{aux}}\}$, where $V_{\text{aux}}$ 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:

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

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 $s^{\text{aux}}_{t,u}$ to obtain the probabilities over the auxiliary label space $\mathcal{Y}_{\text{aux}}$:

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

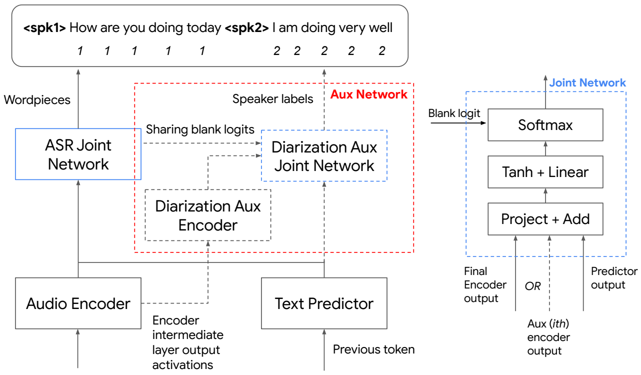

Figure 1: *Diagram of the proposed ASR-diarization multi-task architecture with an auxiliary network of an auxiliary encoder and a separate joint network sharing ASR blank logits. Dotted lines indicate the newly added designs. Solid outlined modules are frozen during training.*

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

### 3.2. Proposed multi-task ASR and diarization system

In the context of speaker diarization, auxiliary labels are now speakers. The synchronization between ASR and diarization naturally enables the word-level speaker diarization task. Therefore, we further extend the multi-output 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 [33], 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, 34], 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 $A^{\text{aux}}$ and $b_s^{\text{aux}}$.

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 |

Table 2: *ASR and diarization performance of the baseline and proposed models. WERs (%) are reported with 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 |
| 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 |

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

| 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 |

# 4. Experiments

## 4.1. Datasets

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

1. AMI [35]: 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.
2. Callhome [36]: we use the Callhome American English Speech official training and evaluation subsets.
3. Fisher [37]: we withhold a test subset of 172 utterances[2] from Fisher English Training Speech and use the rest for training.

The training data are segmented into 15, 30 and 60 second segments. The segmentation avoids chopping in the middle of a sentence. For the test splits, we evaluate the original full-length audios and 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 [38] 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 $0 \sim 2$ utterances for variety. Remaining samples are concatenated in random order, with inserted pause ($0.2 \sim 1.5$ seconds) and cross-fade ($0 \sim 0.2$ seconds). 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 testsets, we sample from test-clean and test-other.

## 4.2. Model architecture

We extract 128-dimensional log-Mel filterbank features using a 32ms Hamming window with a hop length of 10ms. Frames are stacked by 4 and sub-sampled by 3 to generate 512-dimensional features at 30ms frame rate. The causal ASR encoder consists of 12 conformer layers [39] of 512 dimensions with funnel pooling [40] and outputs $D_a = 512$ feature dimensions. The embedding-based decoder computes language model features of $D_l = 640$ dimensions, based on two previous non-blank tokens [41]. The ASR joint network hidden dimension is $D_h = 640$. A last linear layer projects the hidden dimension to the wordpiece model vocabulary size $V = 4096$.

The 5th ASR Conformer layer outputs are fed into our auxiliary network, which has 9 LSTM layers, each layer with 1024 hidden nodes and 512 output nodes. The auxiliary joint has 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 $N = 1$ to 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 a pre-trained ASR model [41] and freeze the parameters of 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 use the speaker label RNN-T loss (with the shared blank logits) because ASR is frozen.

## 4.3. Baseline: turn-based diarization

We set up the turn-based diarization baseline following the "turn-to-diarize" system [2, 42] without pairwise speaker turn constraints and apply multi-stage clustering [43]. 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. We choose WDER over other metrics like cpWER [25] or Diarization Error Rate (DER). WDER is more decoupled from ASR quality than cpWER. DER is not applicable in our problem because word-level end-to-end systems do not involve word timings.

## 4.5. Experimental results

ASR and diarization quality is reported in Table 2. We noticed the high WER of the ASR model, due to various reasons including: Callhome's low quality ground truth; ASR not trained with overlapping speech; non-standard annotations (e.g. "mhm", "y-ye- yes"); and domain mismatch (ASR not trained on read speech like LibriSpeech). For diarization, our model outperforms the baseline on 5-min Callhome test data by 25%. On simulated test data, our model shows the capability to diarize multi-speaker ut-

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

Table 4: *Impact of intermediate layer selection on WDER (%). Callhome is abbreviated as CH.*

| Intermediate Layer Selection | CH | Fisher | Sim |
|---|---|---|---|
| 0th Conf layer (features) | 23.8 | 24.5 | 10.4 |
| 5th Conf layer (proposed) | 7.7 | 8.0 | 4.3 |
| 12th Conf layer (last) | 33.6 | 37.3 | 46.9 |

Table 5: *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 |
|---|---|---|---|---|---|
| Proposed | 7.7 | 9.0 | 8.0 | 3.7 | 4.3 |
| -Simulated | 11.6 | 9.8 | 12.3 | 5.4 | 22.2 |
| -30/60s segments | 28.8 | 22.5 | 22.1 | 15.7 | 26.8 |

terances, up to 4 speakers over 90 seconds. However, we suspect the performance degrades on longer audio duration, as shown by the 10-min Fisher test data evaluation result.

Therefore, we take a deeper look into the short-form performance on segmented test utterances in Table 3. Our proposed model outperforms the baseline on short-form Callhome and Fisher significantly across various segment lengths, particularly on 30s segments by 32% and 56%. The performance does degrade if the audio gets very long (e.g. compared against 10-min Fisher in Table 2). The long-form degradation of the proposed system stems from the mismatch between training segments and the full length test segments, and can be addressed by carrying cross-segment speaker information to the training process. Meanwhile, the baseline system performs better on longer test segments (monotonically decreasing WDER on Fisher and Callhome, except for its 60s metric). This observation is consistent with previous studies on the clustering-based baseline system [43]: clustering algorithms usually perform reasonably well on very long utterances, but suffer from short segments due to insufficient samples.

### 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 4, neither the first layer nor the last layer works well. This aligns with our expectation and conclusions from previous work [19, 34]. 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 5 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 and suggests that we should include longer training segments if feasible.

Table 6: *Pre-segmented short-form AMI WDER (%), breakdown by reference number of speakers. AMI has 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, we report a modified WDER[3] which does not count words that overlap with any other word in the ground truth.*

| AMI | Baseline WDER (%) | | | | Proposed WDER (%) | | | |
|---|---|---|---|---|---|---|---|---|
| Lengths | 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 | -[4] | 6.4 | 4.4 | 9.3 | - | 9.8 | 15.8 | 20.8 |

### 4.7. Limitations

On multi-speaker, 30-minute AMI test data, we observed that the proposed model did not work. To investigate further, we break down the AMI WDER by the number of speakers in Table 6. Our model performs better than the baseline on 1-speaker and 2-speaker scenarios up to 60 seconds of audio duration. Diarization quality starts to drop when the audio gets longer, or when the number of speakers increases. Based on this and other experimental results, we summarize the limitations of our model.

Firstly, a sufficient amount of in-domain training data is crucial for model performance. Even though out-of-domain data helps to some extent (as shown by the simulated data in Table 5), with only 80 hours of AMI training data, the model still can not learn to diarize AMI competently in our experiments. This is the data domain mismatch. On the other hand, full-length utterances (e.g. 1-hour audio) can not be directly fed into training batches because of hardware memory constraints. Training data is typically sliced into shorter segments, while test data can be of arbitrary length and extremely long. This leads to the audio duration mismatch between training and inference.

## 5. Conclusions

This paper explored word-level end-to-end neural diarization via auxiliary networks, a framework that allows adding speaker diarization capability to any off-the-shelf RNN-T based ASR model, without WER regression. It is also a simpler architecture, with less privacy risk, as it does not involve enrollment, speaker profiles, clustering, or turn-based segmentation. We observed superior performance over our conventional modularized system on 2-speaker short-form scenarios, even with an out-of-domain frozen ASR model. Compromised quality was observed on 3+speaker and long-form scenarios due to data domain and duration mismatches in the training data. To address this, future work would include developing advanced data augmentation techniques to simulate large amounts of in and out of domain conditions with arbitrary number of speakers, as well as cross-segment training with historical context for long-form generalization. Lastly, this architecture can be further extended to serialized output training (SOT) for overlapping speech.

[3] `https://github.com/google/speaker-id/tree/master/publications/WEEND#wder-and-modified-wder-without-overlapping-words`

[4] For the evaluation on 120-sec segments, since there are only 6 single speaker test examples, we do not list these results.

# 6. References

[1] I. Medennikov *et al.*, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," in *Interspeech*, 2020.

[2] W. Xia *et al.*, "Turn-to-Diarize: Online speaker diarization constrained by transformer transducer speaker turn detection," in *ICASSP*. IEEE, 2022, pp. 8077–8081.

[3] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, "ECAPA-TDNN embeddings for speaker diarization," *arXiv preprint arXiv:2104.01466*, 2021.

[4] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. Lopez Moreno, "Speaker diarization with LSTM," in *ICASSP*. IEEE, 2018, pp. 5239–5243.

[5] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *Interspeech*, 2017, pp. 2739–2743.

[6] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *ICASSP*. IEEE, 2017, pp. 4930–4934.

[7] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *ICASSP*. IEEE, 2019, pp. 6301–6305.

[8] Q. Li, F. L. Kreyssig, C. Zhang, and P. C. Woodland, "Discriminative neural clustering for speaker diarisation," in *SLT*. IEEE, 2021.

[9] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Interspeech*, 2019, pp. 4300–4304.

[10] S. 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] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, J. Shi, and K. Nagamatsu, "Neural speaker diarization with speaker-wise chain rule," *arXiv preprint arXiv:2006.01796*, 2020.

[12] Y. Xue, S. Horiguchi, Y. Fujita, S. Watanabe, P. García, and K. Nagamatsu, "Online end-to-end neural diarization with speaker-tracing buffer," in *SLT*. IEEE, 2021, pp. 841–848.

[13] E. Han, C. Lee, and A. Stolcke, "Bw-eda-eend: Streaming end-to-end neural speaker diarization for a variable number of speakers," in *ICASSP*. IEEE, 2021, pp. 7193–7197.

[14] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *ICASSP*. IEEE, 2021, pp. 7198–7202.

[15] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[16] C. Zhang and Q. Wang, "Speaker diarization: A journey from unsupervised to supervised approaches," Odyssey: The Speaker and Language Recognition Workshop, 2022, tutorial session.

[17] W. Wang, D. Zhao, S. Ding, H. Zhang, S.-Y. Chang, D. Rybach, T. N. Sainath, Y. He, I. McGraw, and S. Kumar, "Multi-output RNN-T joint networks for multi-task learning of ASR and auxiliary tasks," in *ICASSP*. IEEE, 2023, pp. 1–5.

[18] C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig, "Improving RNN transducer based ASR with auxiliary tasks," in *SLT*. IEEE, 2021, pp. 172–179.

[19] J. Zhang, Y. Peng, P. Van Tung, H. Xu, H. Huang, and E. S. Chng, "E2E-based multi-task learning approach to joint speech and accent recognition," *arXiv preprint arXiv:2106.08211*, 2021.

[20] L. E. Shafey, H. Soltau, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," in *Interspeech*, 2019, pp. 396–400.

[21] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers," *arXiv preprint arXiv:2006.10930*, 2020.

[22] N. Kanda, Z. Meng, L. Lu, Y. Gaur, X. Wang, Z. Chen, and T. Yoshioka, "Minimum bayes risk training for end-to-end speaker-attributed ASR," in *ICASSP*. IEEE, 2021, pp. 6503–6507.

[23] N. Kanda, G. Ye, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "End-to-end speaker-attributed ASR with transformer," *arXiv preprint arXiv:2104.02128*, 2021.

[24] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, "Streaming speaker-attributed ASR with token-level speaker embeddings," *arXiv preprint arXiv:2203.16685*, 2022.

[25] N. Kanda, X. Chang, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Investigation of end-to-end speaker-attributed ASR for continuous multi-talker recordings," in *SLT*. IEEE, 2021, pp. 809–816.

[26] K. Zmolikova *et al.*, "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures." in *Interspeech*, 2017, pp. 2655–2659.

[27] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with speaker beam," in *ICASSP*. IEEE, 2018, pp. 5554–5558.

[28] M. Delcroix, S. Watanabe, T. Ochiai, K. Kinoshita, S. Karita, A. Ogawa, and T. Nakatani, "End-to-end speakerbeam for single channel target speech recognition." in *Interspeech*, 2019, pp. 451–455.

[29] N. Kanda, S. Horiguchi, R. Takashima, Y. Fujita, K. Nagamatsu, and S. Watanabe, "Auxiliary interference speaker loss for target-speaker speech recognition," *arXiv preprint arXiv:1906.10876*, 2019.

[30] S. De Silva, A. Liu, and L. Nabarro, "Europe's tough new law on biometrics," *Biometric Technology Today*, vol. 2017, no. 2, pp. 5–7, 2017.

[31] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv:1211.3711*, 2012.

[32] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (HAT)," in *ICASSP*, 2020.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] J. Zhang *et al.*, "Intermediate-layer output regularization for attention-based speech recognition with shared decoder," *arXiv preprint arXiv:2207.04177*, 2022.

[35] J. Carletta *et al.*, "The AMI meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*. Springer, 2005, pp. 28–39.

[36] A. Canavan, D. Graff, and G. Zipperlen, "CALLHOME American English speech LDC97S42," LDC Catalog. Philadelphia: Linguistic Data Consortium, 1997.

[37] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: A resource for the next generations of speech-to-text," in *LREC*, vol. 4, 2004, pp. 69–71.

[38] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[39] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.

[40] Z. Dai, G. Lai, Y. Yang, and Q. Le, "Funnel-transformer: Filtering out sequential redundancy for efficient language processing," *Advances in Neural Information Processing Systems*, 2020.

[41] R. Botros, T. Sainath, R. David, E. Guzman, W. Li, and Y. He, "Tied & reduced RNN-T decoder," in *Interspeech*, 2021.

[42] G. Zhao *et al.*, "Augmenting transformer-transducer based speaker change detection with token-level training loss," in *Proc. ICASSP*, 2023.

[43] Q. Wang, Y. Huang, H. Lu, G. Zhao, and I. L. Moreno, "Highly efficient real-time streaming and fully on-device speaker diarization with multi-stage clustering," *arXiv preprint arXiv:2210.13690*, 2022.