Streaming Multi-Talker ASR with Token-Level Serialized Output Training

Naoyuki Kanda¹, Jian Wu¹, Yu Wu², Xiong Xiao¹, Zhong Meng¹, Xiaofei Wang¹, Yashesh Gaur¹, Zhuo Chen¹, Jinyu Li², Takuya Yoshioka¹

¹Microsoft Cloud+AI, USA  ²Microsoft Research Asia, China

{nakanda,wujian,yuwul,xioxiao,zhme,xiaofewa,yagaur,zhuc,jinyli,tayoshio}@microsoft.com

Abstract

This paper proposes a token-level serialized output training (t-SOT), a novel framework for streaming multi-talker automatic speech recognition (ASR). Unlike existing streaming multi-talker ASR models using multiple output branches, the t-SOT model has only a single output branch that generates recognition tokens (e.g., words, subwords) of multiple speakers in chronological order based on their emission times. A special token that indicates the change of “virtual” output channels is introduced to keep track of the overlapping utterances. Compared to the prior streaming multi-talker ASR models, the t-SOT model has the advantages of less inference cost and a simpler model architecture. Moreover, in our experiments with LibriSpeechMix and LibriCSS datasets, the t-SOT-based transformer transducer model achieves the state-of-the-art word error rate (WER) for various multi-talker test sets including overlapping speakers (e.g., [10, 11, 12, 13, 14]), where the model is often trained with permutation invariant training [15, 16, 17]. Another recently proposed approach is serialized output training (SOT) [18], which uses a model that has only a single output branch. In SOT, the single output branch generates transcriptions for multiple speakers one after another, where the speaker-wise transcriptions are interleaved by a special separator token that indicates the speaker change. The SOT-based ASR model achieved the state-of-the-art (SOTA) word error rate (WER) [19, 20, 21] for various multi-talker test sets including LibriSpeechMix [18], LibriCSS [6] and AMI [22]. However, the SOT model assumes the attention-based encoder-decoder (AED) [23, 24] as a backbone ASR system, which renders the model usable only for the offline (i.e. non-streaming) inference. A few recent studies explored the streaming multi-talker ASR problem to transcribe each spoken word with a low latency even for overlapping speech. Streaming unmixing and recognition transducer (SURT) [25] and multi-speaker recurrent neural network transducer (MS-RNN-T) [26] were concurrently proposed based on a similar idea, where the model has two output branches to generate two simultaneous transcriptions for overlapping speech. However, their reported WERs still lagged far behind the SOTA result of the offline SOT-model.

In this paper, we present token-level serialized output training (t-SOT), a novel streaming multi-talker ASR framework. With t-SOT, words spoken by overlapping speakers are generated by a single output branch in a chronological order based on their emission times.³ A special token indicating “virtual” output channels is used to keep track of transcriptions for overlapping speakers.² Compared to the prior streaming multi-talker ASR models [25, 26], the t-SOT model has advantages of less inference cost and a simpler model architecture. Moreover, our experimental results with LibriSpeechMix and LibriCSS datasets show that the transformer transducer (TT) [31] trained with t-SOT framework achieves new SOTA results by a significant margin to the prior results including the offline SOT-model.

2. Streaming multi-talker ASR with t-SOT

2.1. Basic t-SOT for up to two concurrent utterances

We first explain the basic idea of t-SOT by assuming that the input audio contains overlapping speech of up to two concurrent

²The idea of emission-time-based serialization was concurrently proposed in [27]. However, their speaker tracking method is different from ours, and their experiments were limited to non-streaming models.

³While several prior works proposed to insert special tokens in a transcription [28, 29, 30, 18], our work is the first to propose a special token for transcribing overlapping utterances in a streaming fashion.
Table 1: Comparison of representative multi-talker ASR frameworks. A preferable property in each row is presented in bold font.

| Feature                                | SOT [18] | SURT [25], MS-RNN-T [26] | t-SOT (proposed method) |
|----------------------------------------|----------|--------------------------|-------------------------|
| Streaming inference                    | Not-available | Available | Available |
| Inference cost on decoder              | Same with single-talker model | K-times* of single-talker model | Same with single-talker model |
| Max concurrent utterances              | Unlimited | Pre-defined | Pre-defined |
| Speaker counting                       | Available | Not-available | Not-available |
| Model architecture                     | Same with single-talker model | Multiple output branches | Same with single-talker model |
| Restriction on ASR-type                | Restricted to AED-based ASR | No restriction | No restriction |
| Accuracy on non-overlapping audio°    | Good     | Bad          | Good          |
| Accuracy on overlapping audio°         | Good     | Fair         | Good          |

Algorithm 1: Generating a serialized transcription for a training sample with up to two concurrent utterances

1: Given a speech sample with a time- and speaker-annotated transcription $T = \langle w_i, e_i, s_i \rangle_{i=1}^{N}$, where $w_i$ is the $i$-th token, and $e_i$ and $s_i$ are the emission time and the speaker of $w_i$, respectively.
2: Sort $T$ in ascending order of $e_i$.
3: Initialize a list of tokens $W \leftarrow \{w_1\}$.
4: for $i = 2, ..., N$ do
5: if $s_i \neq s_{i-1}$ then
6: Append $(cc)$ to $W$.
7: Append $w_i$ to $W$.
8: return $W$

Table 1 summarizes the differences between t-SOT and representative multi-talker ASR methods. Compared with SOT [18] which can be used only for non-streaming inference, a t-SOT model can recognize each spoken word in a streaming fashion. The t-SOT model is also advantageous because the model architecture is not restricted to the AED. On the other hand, t-SOT has the limitation on the maximum number of concurrent utterances that the model can recognize. The t-SOT model also cannot count the number of speakers in the audio segment.

2.3. Comparison to prior works

For streaming multi-talker ASR, the t-SOT framework has various advantages over SURT [25] and MS-RNN-T [26]. Firstly, t-SOT requires only a single decoding process as with the conventional single-talker ASR while SURT and MS-RNN-T require to execute the decoder multiple times (i.e., one decoder run for each output branch). Therefore, the t-SOT model fundamentally requires less computation than the prior models. Secondly, the t-SOT model architecture is much simpler than SURT and MS-RNN-T because the same architecture as the single-talker ASR model can be used without any modification. Thirdly, our experimental results showed that the t-SOT-based ASR model achieved significantly better WER than the prior SOTA results for both non-overlapping and overlapping speech as detailed in the next section.

3. Experiments

We first conducted an experiment by using the LibriSpeechMix evaluation set [18], where we limited the number of speakers in each audio segment to up to two. We then conducted an evaluation with LibriCSS [6] where each long-form audio contains many utterances from 8 speakers.

3.1. Experiment with LibriSpeechMix

3.1.1. Experimental settings

As the first evaluation dataset, we used LibriSpeechMix [18], which is made by mixing up to three utterances randomly sampled from LibriSpeech [39]. In this work, we used the single-speaker (i.e. non-overlapping) evaluation set and the two-speaker-mixed evaluation set. For the two-speaker-mixed evaluation set, two utterances are mixed with a randomly-determined delay such that each evaluation sample contains a partial speaker overlap. We evaluated the WER in the same way as the prior work [18] did. That is, the recognition hypotheses (= one or two hypotheses after deserialization) and the
Table 2: WER (%) for LibriSpeechMix test set with various non-streaming and streaming multi-talker ASR models. The algorithmic latency is shown in the “Latency” column, where 160 msec is our target configuration. No language model was used for all results including the ones in prior works.

| Model                  | # of param. | Latency (msec) | Test WER |
|------------------------|-------------|----------------|----------|
| (Non-streaming Models) |             |                |          |
| PIT LSTM-AED [18]      | 161M        | ∞              | 6.7 11.9 |
| SOT LSTM-AED [35]      | 136M        | 4.5           | 10.3     |
| SOT Conformer-AED [19] | 129M        | ∞              | 3.6 4.9  |

| (Streaming Models)     |             |                |          |
|------------------------|-------------|----------------|----------|
| LSTM-SURT [36]         | 81M         | 150            | -        |
| PIT-MS-RNN-T [26]      | 81M         | 30             | 7.6 10.2 |
| Transformer-SURT [37]  | 85M         | 1000           | -        |
| Transformer-SURT-T [38] | 81M        | 30             | -        |
| t-SOT TT-18            | 82M         | 40             | 5.1 8.4  |
| t-SOT TT-18            | 82M         | 160            | 4.9 6.9  |
| t-SOT TT-18            | 82M         | 640            | 4.2 6.2  |
| t-SOT TT-18            | 82M         | 1250           | 3.9 5.2  |
| t-SOT TT-36            | 139M        | 160            | 4.3 6.2  |
| t-SOT TT-36            | 139M        | 2500           | 3.3 4.4  |

† Models with two output branches, for which most parameters are loaded and used twice in the inference.

references are compared for all possible speaker permutations, and the speaker permutation that produces the minimum number of errors is selected to calculate the WER. A hypothesis (or reference) that does not have a corresponding reference (or hypothesis) is regarded as all insertion (or deletion) errors.

We simulated the training data by randomly mixing up to two utterances from “train,960” of the LibriSpeech training data. In the training data generation, for each sample, we first selected the number of speakers $S$ from $\{1, 2\}$ with $p\%$ and $(100 - p)\%$ probabilities for $S = 1$ and 2, respectively, where $p = 50$ unless otherwise stated. We then randomly selected $S$ utterances from “train,960”. If $S$ was equal to two, we further randomly sampled the delay $d$ for the second utterance $u_2$ from Uniform$(0, \text{len}(u_1))$, where len$(u_1)$ is the duration of the first utterance $u_1$, Utterance $u_1$ and utterance $u_2$, delayed by $d$, were then mixed without changing the original volumes. We used the time-alignment generated by the Montreal Forced Aligner [40] to generate the serialized transcription for each training sample. To increase the variability of the training data, we applied the speed perturbation [41] with the ratios of $[0.9, 1.0, 1.1]$, the volume perturbation with the ratio between 0.125 to 2.0, and the adaptive SpecAugment [42]. Following [20, 38], we simulated the training data on the fly to generate infinite variations of the training samples.

For the streaming ASR model, we used a TT with a chunk-wise look-ahead proposed in [43]. The encoder consists of 2 convolution layers, each of which halves the time resolution, followed by a 18-layer or 36-layer transformer with relative positional encoding. We refer to the model with the 18-layer transformer as “TT-18” and the model with the 36-layer transformer as “TT-36”. Each transformer block consists of a 512-dim multi-head attention with 8 heads and a 2048-dim point-wise feed forward layer with Gaussian error linear unit (GELU) activation function. Our TT’s prediction network consists of 2 layers of 1024-dim long short-term memory (LSTM). We used 4,000 word pieces plus blank and (|c|) tokens as the recognition units. The audio input feature is an 80-dim log mel-filterbank.

Table 3: WER (%) on LibriSpeechMix for single- and multi-talker models with different training configurations. All models have 160 msec of algorithmic latency.

| Model                  | Training data | Dev WER | Test WER |
|------------------------|---------------|---------|----------|
|                         | 1spk          | 2spk    | 1spk     | 2spk     |
| single-talker TT-18     | 100%          | -       | 4.3 63.2 | 4.5 63.7 |
| t-SOT TT-18            | 67% 33%       | -       | 4.3 7.3  | 4.7 7.4  |
| t-SOT TT-18            | 50% 50%       | -       | 4.3 6.9  | 4.9 6.9  |
| single-talker TT-36     | 100%          | -       | 3.9 6.2  | 4.4 6.3  |
| t-SOT TT-36            | 67% 33%       | -       | 3.8 6.4  | 4.4 6.3  |
| t-SOT TT-36            | 50% 50%       | -       | 3.9 6.0  | 4.3 6.2  |

extracted every 10 msec. As proposed in [43], we controlled the algorithmic latency of the model based on the chunk size of the attention mask. The minimum possible latency is 40 msec, which is determined by the time resolution of the input feature sequence and the 2 convolutional layers with 2-times subsampling in the encoder. For all models, we performed 225K training iterations with 16 GPUs, each of which consumed mini-batches of 12,000 frames. We used an AdamW optimizer with a linear decay learning rate schedule with a peak learning rate of 1.5e-3 after 25K warm up iterations.

3.1.2. Main results

Table 2 shows the comparison of our t-SOT model and prior multi-talker ASR models on the LibriSpeechMix test set. We evaluated t-SOT TT models with various algorithmic latency and model sizes. Firstly, we observed that the t-SOT TT-18 with only 40 msec algorithmic latency already outperformed the results of all prior streaming multi-talker ASR models. Note that even though t-SOT TT-18 has almost the same number of parameters with SURT [25, 36] or MS-RNN-T [26, 38], t-SOT is more time- and space-efficient in the inference because SURT and MS-RNN-T run decoding twice, once for each of the two output branches. Secondly, we observed a significant WER reduction by increasing algorithmic latency and the model size. In our experiment, enlarging the latency for 4 times (e.g., 160 msec to 640 msec) achieved a similar level of WER reduction to doubling the number of layers (i.e., TT-18 to TT-36). Notably, our t-SOT TT-36 with 2,560 msec latency achieved 3.3% and 4.4% of WERs for single-speaker and two-speaker-mixed test sets, respectively, which are even better than the prior SOTA results by the offline SOT Conformer-AED [19].

3.1.3. Comparison with single-talker ASR models

Table 3 shows a comparison of various single-talker and t-SOT models. For the t-SOT model, we tested different mixtures of single- and two-speaker training samples by changing $p$ in the on-the-fly data generation (Section 3.1.1). Firstly, as we expected, the single-talker TT model showed bad WERs for two speaker overlapping speech, which proves the necessity of multi-talker ASR modeling. Secondly, we observed marginal WER improvement for the single-speaker evaluation set when the model observed more single-speaker training samples, which however came with the cost of WER degradation for the two-speaker-mixed evaluation set. Thirdly, we observed a slight degradation in single-talker WER for the t-SOT TT-18 compared to the single-talker TT-18 most likely because TT-18...
Table 4: WER (%) on the monaural LibriCSS test set in the continuous input evaluation setting. A macro average of WERs is shown in the “Avg.” column. 0L and 0S are 0% overlap conditions with long and short inter-utterance silences. For each overlapping condition, the best WER with streaming ASR models is shown in bold font, and the best number among all ASR models is shown with underline.

| System | Algorithmic latency | WER (%) for different overlap ratio |
|--------|---------------------|-------------------------------------|
|        |                     | 0L  | 0S  | 10 | 20 | 30 | 40 | Avg. |
| (Non-streaming ASR models with speech separation) |                     |     |     |    |    |    |    |      |
| BLSTM-CSS + Hybrid ASR [6] | 1.2 sec† + (utterance length)* | 16.3 | 17.6 | 20.9 | 26.1 | 32.6 | 36.1 | 24.9 |
| Conformer-CSS + Transformer-AED-ASR w/ LM [9] | 1.2 sec† + (utterance length)* | 6.1  | 6.9  | 9.1  | 12.5 | 16.7 | 19.3 | 11.8 |
| Conformer-CSS + Transformer-AED-ASR w/ LM [44] | 1.2 sec† + (utterance length)* | 6.4  | 7.5  | 8.4  | 9.4  | 12.4 | 13.2 | 9.6  |
| (Streaming ASR models) |                     |     |     |    |    |    |    |      |
| SURT w/ DP-ISTM [45] | 350 msec | 9.8  | 19.1 | 20.6 | 20.4 | 23.9 | 26.8 | 20.1 |
| SURT w/ DP-Transformer [45] | 350 msec | 9.3  | 21.1 | 21.2 | 25.9 | 28.2 | 31.7 | 22.9 |
| Single-talker TT-18 | 160 msec | 7.0  | 7.3  | 14.0 | 20.9 | 27.9 | 34.3 | 18.6 |
| t-SOT TT-18 (proposed) | 160 msec | 6.5  | 6.7  | 13.1 | 20.4 | 27.0 | 34.0 | 18.0 |
| Single-talker TT-36 | 160 msec | 7.5  | 7.5  | 8.5  | 10.5 | 12.6 | 14.0 | 10.1 |
| t-SOT TT-36 (proposed) | 160 msec | 6.7  | 6.1  | 7.5  | 9.3  | 11.6 | 12.9 | 9.0  |

† Latency incurred by CSS. * Latency incurred by VAD and ASR. The average of utterance lengths in the LibriCSS test set is 7.5 sec.

The evaluation results are shown in Table 4. We observed the proposed t-SOT TT model achieved significantly better WER for all conditions compared to the prior streaming ASR models. Surprisingly, t-SOT TT-36 with 160 msec latency even outperformed the strong prior results based on the Conformer-based continuous speech separation (CSS) [9] and an offline Transformer-AED-based ASR [47] with language model (LM) fusion for 5 out of 6 conditions, resulting in the new SOTA average WER. This result strongly indicates the advantage of the end-to-end multi-talker modeling over the approach to combine independent modules. It is also noteworthy that the t-SOT models came close to the same-sized single-talker TT models’ performance for the non-overlapping test conditions (0L and 0S). The TTTT-36 even outperformed the single-talker TT-36 for non-overlapping conditions on average (6.4% vs. 6.6%), which is consistent with the result for LibriSpeechMix (Section 3.1.3).

4. Conclusions

In this paper, we presented t-SOT, a novel framework for streaming multi-talker ASR. Unlike prior streaming multi-talker ASR models, the t-SOT model has only a single output branch that generates recognition tokens from overlapping speakers in chronological order based on their emission times. A special token that indicates the change of the virtual output channels is introduced to keep track of the overlapping speakers. We evaluated t-SOT with LibriSpeechMix and LibriCSS and showed that t-SOT model achieved new SOTA results even with streaming inference using a simpler model architecture.

6It is highly computational demanding to calculate SAgWER especially for many hypotheses without time alignment information [46]. Due to the difficulty, we don’t have a comparable result for SOT-based ASR models. For t-SOT models, we were still able to calculate SAgWER since the number of concurrent hypotheses was limited to two.
5. References

[1] Ö. Çetin and E. Shriberg, “Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition,” in Interspeech, 2006.

[2] T. Yoshokura et al., “Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks,” in Proc. Interspeech, 2018, p. 3038–3042.

[3] N. Kanda et al., “Guided source separation meets a strong ASR backend: Hitachi/Paderborn University joint investigation for dinner party ASR,” in Interspeech, 2019, pp. 1248–1252.

[4] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ‘CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” Interspeech, pp. 1561–1565, 2018.

[5] S. Watanabe et al., “CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings,” in CHiME 2020, 2020.

[6] Z. Chen, T. Yoshokura, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, and J. Li, “Continuous speech separation: dataset and analysis,” in ICASSP, 2020, pp. 7284–7288.

[7] D. Raj et al., “Integration of speech separation, diarization, and recognition for multi-speaker systems: System description, comparison, and analysis,” in SLT, 2021, pp. 897–904.

[8] T. Yoshokura et al., “Advances in online audio-visual meeting transcription,” in ASRU, 2019, pp. 276–283.

[9] S. Chen, Y. Wu, Z. Chen, J. Wu, J. Li, T. Yoshokura, C. Wang, S. Liu, and M. Zhou, “Continuous speech separation with Conformer,” in ICASSP, 2021, pp. 5749–5753.

[10] D. Yu, X. Chang, and Y. Qian, “Recognizing multi-talker speech with permutation invariant training,” Interspeech, pp. 2456–2460, 2017.

[11] H. Seki, T. Hori, S. Watanabe, J. Le Roux, and J. R. Hershey, “A purely end-to-end system for multi-speaker speech recognition,” in ACL, 2018, pp. 2620–2630.

[12] X. Chang, Y. Qian, K. Yu, and S. Watanabe, “End-to-end monaural multi-speaker ASR system without pretraining,” in ICASSP, 2019, pp. 6256–6260.

[13] X. Chang et al., “MIMO-SPEECH: End-to-end multi-channel multi-speaker speech recognition,” in ASRU, 2019, pp. 237–244.

[14] A. Tripathi, H. Lu, and H. Sak, “End-to-end multi-talker overlapping speech recognition,” in ICASSP, 2020, pp. 6129–6133.

[15] J. R. Hershey et al., “Deep clustering: Discriminative embeddings for segmentation and separation,” in ICASSP, 2016, pp. 31–35.

[16] Y. Isik, J. L. Roux, Z. Chen, S. Watanabe, and J. R. Hershey, “Single-channel multi-speaker separation using deep clustering,” in Interspeech, 2016, pp. 545–549.

[17] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in ICASSP, 2017, pp. 241–245.

[18] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshokura, “Seri-alized output training for end-to-end overlapped speech recogni-tion,” in Interspeech, 2020, pp. 2797–2801.

[19] N. Kanda et al., “End-to-end speaker-attribute ASR with Trans-fomer,” in Interspeech, 2021, pp. 4413–4417.

[20] N. Kanda et al., “Large-scale pre-training of end-to-end multi-talker ASR for meeting transcription with single distant micro-phone,” in Interspeech, 2021, pp. 3430–3434.

[21] N. Kanda et al., “A comparative study of modular and joint ap-proaches for speaker-attribute asr on monaural long-form audio,” in ASRU, 2021.

[22] J. Carletta et al., “The AMI meeting corpus: A pre-announcement,” in International workshop on machine learning for multimodal interaction, 2006, pp. 28–39.

[23] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, “End-to-end continuous speech recognition using attention-based recurrent NN: First results,” in NIPS Workshop on Deep Learning, 2014.

[24] J. K. Choworksi et al., “Attention-based models for speech recognition,” in NIPS, 2015, pp. 577–585.

[25] L. Lyu, N. Kanda, J. Li, and Y. Gong, “Streaming end-to-end multi-talker speech recognition,” IEEE Signal Processing Letters, vol. 28, pp. 803–807, 2021.

[26] I. Sklyar, A. Piunova, and Y. Liu, “Streaming multi-speaker ASR with RNN-T,” in ICASSP, 2021, pp. 6903–6907.

[27] X. Chang, N. Moritz, T. Hori, S. Watanabe, and J. Le Roux, “Extended graph temporal classification for multi-speaker end-to-end ASR,” arXiv preprint arXiv:2203.00252, 2022.

[28] S. Watanabe, T. Hori, and J. R. Hershey, “Language independent end-to-end architecture for joint language identification and speech recognition,” in ASRU, 2017, pp. 265–271.

[29] H. Inaguma et al., “An end-to-end approach to joint social signal detection and automatic speech recognition,” in Proc. ICASSP, 2018, pp. 6214–6218.

[30] L. El Shafey, H. Soltani, and I. Shafirn, “Joint speech recogni-tion and speaker diarization via sequence transduction,” in Inter-speech, 2019, pp. 396–400.

[31] Q. Zhang et al., “Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss,” in Proc. ICASSP, 2020, pp. 7829–7833.

[32] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proc. ICML, 2006, pp. 369–376.

[33] A. Graves, “Sequence transduction with recurrent neural networks,” arXiv preprint arXiv:1211.3711, 2012.

[34] N. Kanda et al., “Streaming multi-talker ASR with token-level se-rialized output training,” arXiv preprint arXiv:2202.00842, 2022.

[35] N. Kanda et al., “Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers,” in Interspeech, 2020, pp. 36–40.

[36] L. Lu, N. Kanda, J. Li, and Y. Gong, “Streaming multi-talker speech recognition with joint speaker identification,” arXiv preprint arXiv:2104.02109, 2021.

[37] L. Lu, J. Li, and Y. Gong, “Endpoint detection for streaming end-to-end multi-talker ASR,” arXiv preprint arXiv:2201.09979, 2022.

[38] I. Sklyar, A. Piunova, X. Zheng, and Y. Liu, “Multi-turn RNN-T for streaming recognition of multi-party speech,” arXiv preprint arXiv:2112.10200, 2021.

[39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in ICASSP, 2015, pp. 5206–5210.

[40] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Son-deregger, “Montreal forced aligner: Trainable text-speech align-ment using Kaldi,” in Interspeech, 2017, pp. 498–502.

[41] T. Ko et al., “Audio augmentation for speech recognition,” in In-terspeech, 2015, pp. 3586–3589.

[42] D. S. Park et al., “Specaugment on large scale datasets,” in Proc. ICASSP, 2020, pp. 6879–6883.

[43] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, “Developing real-time streaming transformer transducer for speech recognition on large-scale dataset,” in Proc. ICASSP, 2021, pp. 5904–5908.

[44] J. Wu et al., “Investigation of practical aspects of single channel speech separation for asr,” arXiv preprint arXiv:2107.01922, 2021.

[45] D. Raj, L. Lu, Z. Chen, Y. Gaur, and J. Li, “Continuous stream-ing multi-talker ASR with dual-path transducers,” arXiv preprint arXiv:2109.08555, 2021.

[46] J. G. Fiscus, J. Ajot, N. Radde, and C. Laprun, “Multiple dimen-sion Levenshtein edit distance calculations for evaluating auto-matic speech recognition multi-channel recognition systems during simultaneous speech,” in LREC, 2006, pp. 803–808.

[47] C. Wang et al., “Semantic mask for transformer based end-to-end speech recognition,” in Interspeech, 2020, pp. 971–975.