Separator-Transducer-Segmenter: Streaming Recognition and Segmentation of Multi-party Speech

Ilya Sklyar, Anna Piunova, Christian Osendorfer

Abstract

Streaming recognition and segmentation of multi-party conversations with overlapping speech is crucial for the next generation of voice assistant applications. In this work we address its challenges discovered in the previous work on multi-turn recurrent neural network transducer (MT-RNN-T) with a novel approach, separator-transducer-segmenter (STS), that enables tighter integration of speech separation, recognition and segmentation in a single model. First, we propose a new segmentation modeling strategy through start-of-turn and end-of-turn tokens that improves segmentation without recognition accuracy degradation. Second, we further improve both speech recognition and segmentation accuracy through an emission regularization method, FastEmit, and multi-task training with speech activity information as an additional training signal. Third, we experiment with end-of-turn emission latency penalty to improve end-point detection for each speaker turn. Finally, we establish a novel framework for segmentation analysis of multi-party conversations through emission latency metrics. With our best model, we report 4.6% abs. turn counting accuracy improvement and 17% rel. word error rate (WER) improvement on LibriCSS dataset compared to the previously published work.

Index Terms: streaming multi-speaker speech recognition, speech segmentation, separator-transducer-segmenter

1. Introduction

Automatic speech recognition (ASR) of multi-party recordings with overlapping speech has posed a major scientific challenge for many decades [1, 2, 3, 4]. While single-speaker ASR became ubiquitous through applications like voice assistants (Amazon Alexa, Google Home, etc.), its current capability is limited to scenarios with one active speaker at a time. Apart from ASR, speech overlaps also introduce additional challenges to other parts of the traditional speech processing pipeline as speech segmentation and speaker diarization [5].

Multi-speaker ASR problem was attacked before with both independently optimized modules such as speech separation and speech recognition [6, 7, 8, 9, 10], and jointly optimized multi-speaker end-to-end ASR systems [11, 12, 13, 14, 15, 16]. In [17] a joint multi-speaker ASR and speaker change detection system was proposed to tackle speech recognition and segmentation problems simultaneously in the presence of overlapping speech from arbitrary number of speakers. Follow-up work on serialized output training (SOT) [18, 19, 20, 21, 22] extended it to speaker-attributed ASR that can transcribe “who spoke what” in real multi-speaker conversations with a single integrated model.

In parallel, researchers also investigated multi-speaker ASR performance under streaming conditions, which is crucial for applications with minimal latency. In [23] and [24], two conceptually similar streaming multi-speaker ASR systems, multi-speaker recurrent neural network transducer (MS-RNN-T) and streaming unmixing and recognition transducer (SURT) were proposed simultaneously to enable time-synchronous decoding of partially overlapping speech from 2 speakers. Both models are based on recurrent neural network transducer (RNN-T) model [25] with implicit speech separation in the encoder and multiple decoding threads (one for each speaker). These approaches were later extended to multi-turn audio processing for any number of speakers in [26] and [27], respectively. SURT was additionally extended to perform speaker identification in [28] and endpoint detection in [29], with limitation to 2-speaker single-turn audio recordings. Recently, an alternative approach to streaming multi-speaker ASR, token-level serialized output training (t-SOT), was proposed in [30], which unified single-speaker and multi-speaker model architectures by mixing tokens from all speakers in one sequence and sorting them by the order of their appearance in the audio.

As reported in [26], a naive inclusion of change-of-turn (cot) segmentation tag into a streaming multi-speaker ASR model results in a severe underestimation of the real number of turns in the audio. In this work we attack this issue by explicitly modeling turn boundary with start-of-turn (sot) and end-of-turn (eot) tokens in a novel separator-transducer-segmenter (STS) model. We further improve recognition and segmentation accuracy of this model through FastEmit [31] and masking loss that penalizes leakage of acoustic information into encoder outputs corresponding to non-active speech frames. Finally, we perform segmentation analysis of STS using start-pointing, end-pointing, first subword and last subword emission latency met-

Figure 1: Separator-Transducer-Segmenter. <sot> and <eot> represent start-of-turn and end-of-turn tokens. Model blocks with the same colour have tied parameters, transcripts in the colour-matched boxes belong to the same speaker.
rics, and apply end-of-turn emission latency penalty to regularize end-pointing emission latency.

2. Separator-Transducer-Segmenter

2.1. Separator-Transducer

STS model inherits its separator and transducer functionalities from a multi-turn RNN-T (MT-RNN-T) [26]. MT-RNN-T extends the standard RNN-T [25] to overlapping speech recognition with multiple output channels \( N \), where \( N \) is a maximum number of simultaneously active speakers in the audio. Such design enforces a switch between output channels at speech overlap only and scales to an arbitrary number of speakers. In this work, we only consider cases with \( N = 2 \), and illustrate the corresponding STS model architecture on Fig. 1.

The encoder of STS has a modular structure containing a mixture encoder (MixEnc), \( N \) separation encoders (SepEnc\( n \)) for each output channel \( n \in \{1, \ldots, N\} \) and a recognition encoder (RecEnc) with shared parameters between output channels. The encoder takes acoustic features \( x \) as input and produces high-level disentangled acoustic representations \( h_n \) as output:

\[
h_n = \text{RecEnc}(\text{SepEnc}_n(\text{MixEnc}(x))).
\]

In order to associate \( h_n \) with prediction network outputs for each label sequence \( y_n \), we employ deterministic assignment training (DAT) method [23], which forces the model to learn its asociation output with the speaker order in the audio. In this case, first separation encoder learns to focus on the very first speaker turn, and the second on the follow-up speaker turn if it exists. As a result, DAT computes RNN-T loss only \( N \) times:

\[
L_{\text{RNN-T}} = -\sum_n \log P(y_n|h_n).
\]

2.2. Segmenter

On top of the Separator-Transducer model responsible for multi-speaker speech recognition, here we propose a novel Segmenter functionality to perform segmentation of multi-speaker hypotheses into single-speaker hypotheses and estimate turn boundaries for each speaker turn. To achieve this goal, we introduce a new segmentation modelling strategy in Section 2.2.1, and explore various regularization methods to enhance its performance in Sections 2.2.2, 2.2.3 and 2.2.4.

2.2.1. Segmentation modeling

In [26] change-of-turn (<cct>) tag was introduced in-between per-turn target transcriptions for each output channel. It was reported that this approach underestimated the number of turns in the audio. In this work, we revisit this design choice and introduce two separate tags for turn segmentation: start-of-turn (<sot>) and end-of-turn (<eot>). This approach enables joint start-of-turn and end-of-turn detection and allows interpretation of emission timestamps of these tokens as turn boundaries. In the future, these timestamps can be used for other tasks like speaker diarization (as in [32]) or endpoint detection (as in [29]).

2.2.2. FastEmit

Since in STS model emission timings of <sot> and <eot> tokens act as turn boundaries for potential future application in downstream tasks, it becomes important to regularize their emission latency. To achieve this goal, we use FastEmit [31], a sequence-level emission regularization method. It encourages predicting non-blank tokens and suppresses blank tokens across the entire sequence based on transducer forward-backward probabilities.

2.2.3. Multi-task training with masking loss

To further regularize segmentation capability of the STS model, during training we expose it to the ground-truth segmentation information that is encoded in speech activity labels. Inspired by the work of [16], we employ L2 masking loss to penalize recognition encoder outputs \( h_n \) in regions with no active speaker. The model is trained in a multi-task fashion by jointly optimizing RNN-T and masking losses:

\[
L = L_{\text{RNN-T}} + \gamma \sum_n \ell_2(h_n \circ m_n),
\]

where \( m_n \) is an inverse binary mask of speech activity for output channel \( n \) and \( \gamma \) is a weight of masking loss. This approach enforces encoder outputs \( h_n \) for each output channel \( n \) to be close to 0 in frames where there is no active speaker turn assigned to this output channel.

2.2.4. End-of-turn latency penalty

Besides FastEmit, we also explore another emission regularization method, dynamic latency penalty, applied specifically to the <eot> token. This approach was originally proposed for end-of-speech token emission latency regularization in [33] and investigated for endpoint detection in multi-speaker ASR in [29]. In our case, for each <eot> token in the target transcription, we apply the following dynamic penalty to the probability of <eot> emission in log domain:

\[
\log P((\text{eot})|x_n) = \max(0, \alpha(t - \tau - t_{\text{end}}))
\]

where \( \alpha \) is a tunable scale of a late <eot> emission penalty, \( \tau \) is a <eot> token frame buffer and \( t_{\text{end}} \) is a ground-truth end-of-turn frame. This penalty increases over time and enforces timely emission of the <eot> token.

3. Experimental setup

3.1. Task description

We perform experiments with STS on LibriCSS dataset proposed in [34] for continuous speech separation. It contains 10 one-hour long audio sessions with LibriSpeech utterances played back in a room to simulate meetings with 8 speakers, and it is divided into 6 partitions. 0S and 0L partitions exclude overlapped speech but contain short (S) and long (L) silence gaps between speaker turns, respectively. The remaining partitions represent different overlap ratios from 10% to 40%: OV10, OV20, OV30 and OV40. We use Session 0 of this dataset as a development set to tune decoding hyper-parameters and select best check points, while the remaining Sessions 1-9 are used to report performance.

Following previous work in [26], we adopt an utterance group evaluation protocol (proposed in [20]) for experiments on this dataset. This evaluation protocol enforces segmentation of the original one-hour long audio sessions into utterance group segments using oracle silence boundary information. It ensures the existence of utterance groups containing only one speaker turn (0S, 0L) and utterance groups containing multiple partially overlapping turns. We are aware of the fact that parallel works on streaming multi-speaker ASR [27, 30] recently adopted an alternative continuous input evaluation protocol from [34], and we plan to address this discrepancy in the future work.

3.2. Training setup

The model topology of STS closely follows the one established in the previous work on MS-RNN-T and MT-RNN-T [23, 26].
### Table 1: WER and turn counting accuracy benchmarking of the STS model variants against the baseline on LibriCSS.

<table>
<thead>
<tr>
<th>Model</th>
<th>Turn counting accuracy [%]</th>
<th>Overall</th>
<th>&gt; 2 turns</th>
<th>0L</th>
<th>0S</th>
<th>OV10</th>
<th>OV20</th>
<th>OV30</th>
<th>OV40</th>
<th>full</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT-RNN-T [26]</td>
<td>85.6</td>
<td>28.0</td>
<td>14.7</td>
<td>14.8</td>
<td>20.7</td>
<td>25.3</td>
<td>33.2</td>
<td>36.4</td>
<td>25.3</td>
<td></td>
</tr>
<tr>
<td>STS</td>
<td>89.0</td>
<td>43.0</td>
<td>14.8</td>
<td>14.7</td>
<td>18.1</td>
<td>24.0</td>
<td>32.6</td>
<td>38.8</td>
<td>25.0</td>
<td></td>
</tr>
<tr>
<td>+FastEmit</td>
<td>90.1</td>
<td>47.5</td>
<td>13.0</td>
<td>13.5</td>
<td>16.0</td>
<td>21.3</td>
<td>29.3</td>
<td>31.3</td>
<td>21.7</td>
<td></td>
</tr>
<tr>
<td>+Masking loss</td>
<td>90.2</td>
<td>50.6</td>
<td>13.0</td>
<td>14.0</td>
<td>15.9</td>
<td>18.8</td>
<td>28.6</td>
<td>30.7</td>
<td>21.1</td>
<td></td>
</tr>
</tbody>
</table>

We use 2 LSTM layers in each recurrent module of the architecture (mixture encoder, 2 separation encoders, recognition encoder, prediction network) with 1024 units in each layer. Layer normalization [35] is performed after each LSTM layer. Output layers in the recognition encoder and the prediction network have 640 units. The joint architecture (mixture encoder, 2 separation encoders, recognition encoder, prediction network) with 1024 units in each layer. Layer normalization [35] is performed after each LSTM layer. Output layers in the recognition encoder and the prediction network have 640 units. The joint architecture (mixture encoder, 2 separation encoders, recognition encoder, prediction network) with 1024 units in each layer.

### 5. Segmentation analysis

#### 5.1. Motivation

Results in Section 4 demonstrate the benefit of the proposed segmentation modeling strategy for the turn counting accuracy. However, in the realistic scenario, we are not only interested in the correct prediction of the number of turns in the audio, but also in the turn boundary estimation, i.e. prediction of start-of-turn and end-of-turn timestamps. End-of-turn timestamps can be used for endpoint detection, i.e. to close a current turn and propagate its transcription to downstream services such as natural language understanding (NLU). Both start-of-turn and end-of-turn timestamps can assist speaker diarization in assigning a speaker label to each turn. Therefore, in this section, we propose a methodology for performing comprehensive segmentation analysis of the STS model.

#### 5.2. Methodology

To better understand token emission behavior of <sot> and <eot> tokens, we extract emission timings for both output channels of the STS model. For each analysis, we pick utterances with > 2 turns as the remaining cases are trivial for a two-output system like ours. Moreover, we only focus on ut-
terences with correctly estimated number of turns. We consider the following emission latency metrics for this analysis:

**End-pointing emission latency (EP EL)** – difference between ground-truth end-of-turn timestamp and emission timing of the `<eot>` token. Last turn is omitted.

**Last subword emission latency (LS EL)** – difference between ground-truth end-of-turn timestamp and emission timing of the last subword token in this turn. Last turn is omitted.

**Start-pointing emission latency (SP EL)** – difference between ground truth start-of-turn timestamp and emission timing of the `<sot>` token. First turn is omitted.

**First subword emission latency (FS EL)** – difference between ground truth start-of-turn timestamp and emission timing of the first subword token in this turn. First turn is omitted.

On Fig. 2 an example emission latency analysis is depicted. It contains STS model output timestamps for each word and special token as well as ground-truth start-of-turn (blue dashed lines) and end-of-turn (red dashed lines) timestamps for all turns taken into consideration. As seen from this example, LS EL sets a lower bound on EP EL, while FS EL sets an upper bound on SP EL. Difference between SP and FS EL also shows how much audio context STS model needs to open the next turn without predicting the first subword token.

5.3. Results

In Table 2 we present the results of the segmentation analysis for the STS model variants. Vanilla STS shows the benefit of start-pointing modeling through prediction of turn boundary tokens. It improved turn counting accuracy by 15% abs. on partially overlapping LibriCSS utterances with > 2 turns and enabled segmentation analysis based on emission latency of these tokens. On top of STS, three additional modeling changes were explored: an emission regularization method FastEmit, a multi-task training approach with speech activity signal and end-of-turn emission latency penalty. The former two combined additionally improved turn counting accuracy by 7.6% abs. on utterances with > 2 turns and overall WER by 16% rel., while the latter significantly improved end-pointing emission latency for most turns at the cost of slight WER degradation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean</th>
<th>p50</th>
<th>p90</th>
<th>Mean</th>
<th>p50</th>
<th>p90</th>
<th>Mean</th>
<th>p50</th>
<th>p90</th>
<th>Mean</th>
<th>p50</th>
<th>p90</th>
</tr>
</thead>
<tbody>
<tr>
<td>STS</td>
<td>1428</td>
<td>1100</td>
<td>2611</td>
<td>267</td>
<td>60</td>
<td>230</td>
<td>712</td>
<td>386</td>
<td>666</td>
<td>907</td>
<td>603</td>
<td>794</td>
</tr>
<tr>
<td>+FastEmit</td>
<td>1509</td>
<td>1100</td>
<td>2793</td>
<td>359</td>
<td>10</td>
<td>192</td>
<td>479</td>
<td>211</td>
<td>572</td>
<td>787</td>
<td>555</td>
<td>728</td>
</tr>
<tr>
<td>+Masking loss</td>
<td>1288</td>
<td>980</td>
<td>2711</td>
<td>74</td>
<td>-1</td>
<td>145</td>
<td>332</td>
<td>263</td>
<td>570</td>
<td>561</td>
<td>730</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Segmentation analysis of the STS model variants on LibriCSS. pX is a X-th percentile of emission latency (EL).
7. References


