Interspeech 2022
18-22 September 2022, Incheon, Korea

Turn-Taking Prediction for Natural Conversational Speech

Shuo-yiin Chang, Bo Li, Tara N. Sainath, Chao Zhang, Trevor Strohman, Qiao Liang, Yanzhang He

Google Inc., U.S.A

{shuoyiin,boboli,tsainath,chaoz,strohman,wildstone,yanzhanghe}@google.com

Abstract

While a streaming voice assistant system has been used in many applications, this system typically focuses on unnatural, one-shot interactions assuming input from a single voice query without hesitation or disfluency. However, a common conversational utterance often involves multiple queries with turn-taking, in addition to disfluencies. These disfluencies include pausing to think, hesitations, word lengthening, filled pauses and repeated phrases. This makes doing speech recognition with conversational speech, including one with multiple queries, a challenging task. To better model the conversational interaction, it is critical to discriminate disfluencies and end of query in order to allow the user to hold the floor for disfluencies while having the system respond as quickly as possible when the user has finished speaking. In this paper, we present a turn-taking predictor built on top of the end-to-end (E2E) speech recognizer. Our best system is obtained by jointly optimizing for ASR task and detecting when the user is paused to think or finished speaking. The proposed approach demonstrates over 97% recall rate and 85% precision rate on predicting true turn-taking with only 100 ms latency on a test set designed with 4 types of disfluencies inserted in conversational utterances.

Index Terms: end-to-end models, conversational speech

1. Introduction

Streaming speech recognition systems have been widely used in many voice interaction applications e.g. voice assistant and dialog systems. To achieve a human-level conversational experience, it is essential to learn interaction patterns that resemble human conversational turn-taking. One of the problems is determining when the user has finished speaking, which is typically referred as endpointing [1, 2, 3, 4, 5, 6]. A typical endpointer model makes a series of binary decisions: to wait further for more speech, or to stop listening. While it has been used in many voice systems, these systems often assume "fluent" one-shot voice commands or search queries, where users know exactly what they want to say beforehand. However, a natural conversational utterance commonly involves disfluencies including pauses to think, hesitations, word lengthening, filled pauses (e.g., 'uh', 'um'), and repeated phrases. The disfluencies introduce long pauses in the utterances which could easily cause ambiguity to the E2E model that the user is done speaking. Thus, modeling disfluencies is critical to ensure natural conversational interaction.

To better model the natural conversational interaction, we propose to build a turn-taking model detecting both when the user pauses to think or when they finish speaking under disfluent natural conversation. It is desirable to respond to users immediately when users have done speaking while allowing users to hold the floor if users are pausing to think. As illustrated Figure 1, the conventional endpointer does not allow users to speak with long pauses or filler words, but instead cuts off the user with an inappropriate response. On the other hand, the proposed turn-taking model generates extra cues of pausing that allows the user speak with disfluencies to achieve a conversational interaction experience.

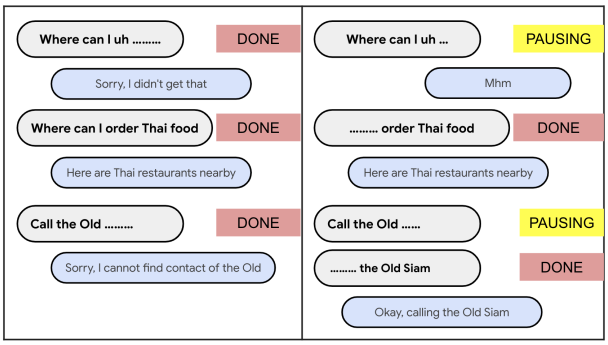

Figure 1: *Conventional endpointer (left) and conversational turn taking model (right)*

Past studies about turn-taking models [7, 8, 9] exploit both acoustic and language model information to predict turn-taking related classes including wait, done speaking or backchanneling e.g. "uh-huh", "yes". In [10, 11], fillers detection is also explored to assist the turn-taking model. The acoustic features e.g., prosody, are investigated in [8, 10] to detect the pauses or a pitch reset at strong phrase boundaries.

The recent development of end-to-end (E2E) models [12, 13, 14, 15, 16] has already shown that having one neural network to do acoustic, pronunciation and language modeling is far better then a modular-based conventional ASR model[17]. Furthermore, we have seen that folding additional detectors into the E2E model, for example the endpointer [4, 5] is far better then having separate modules. Building on this, we propose to build an E2E model that incorporates the turn-taking detector into the E2E model that already folds different components of the speech recognition pipeline into one neural network. This is unlike the previous systems that build an external turn-taking model. The proposed E2E turn-taking detector sees both acoustic representations and intonation patterns from the encoder, as well as grammatically unfinished or finished sentences by decoder, to help aid its performance.

Our best system is obtained by sharing the encoder and the prediction network from the E2E speech recognizer while adapting the joint layers to optimize both the recognition and turn-taking detection. The proposed E2E approach provides 97% recall rate and 85% precision rate on predicting true turn-taking with only 100 ms latency over a test set including 4 types of disfluencies inserted in conversational utterances. The experiments also investigates the acoustic based approach, text based approach and the E2E model on pausing and finishing detection.

10.21437/Interspeech.2022-566

Table 1: *Rules and exceptions for inserting </s> or <pause> annotations.*

When this happens...	It's likely because...	Therefore, insert...
Long silence between words	Speaker finished	<eos>
Silence following last word	Speaker finished	<eos>
Short silence between words	Speaker not finished	<pause>
Silence following lengthened words	Speaker not finished	<pause>
Silence following filler words	Speaker not finished	<pause>
Silence following phrases identified by disfluency detector	Speaker not finished	<pause>

2. Training Data Annotation

To model the conversational turn-taking, the first step is to annotate ground-truth disfluencies in the training data. Specifically, we look to label "user hesitation" as <pause> and "user done speaking" as </s> as an example shown in Fig. 2.

Our training data consists of (1) short-form voice search queries from Google's voice search product, actions on various platforms and (2) long-form data from YouTube, which consists of multiple spontaneous speech sentences including more natural and free voice inputs which include disfluencies. All the utterances are anonymized and hand-transcribed. However, labeling the ground-truth with <pause> and </s> is not straightforward as these labels are not annotated. To address the problem, we performs labeling based on acoustic clues and a text-based disfluency detector as summarized in Table 1.

Specifically, we first insert an </s> when there is a long silence or at the end of the utterance and a <pause> for short silences where the silence segments are obtained by forced alignment. However, disfluenies may also lead to long silence suffix that could be incorrectly labeled as </s> .

To eliminate common mis-insertions, we relabel the silence suffix of word lengthening, filled pauses and other disfluency words as <pause> . To do this, we approximate word lengthening by looking at phoneme duration. The means and standard deviations of phonemes are computed based on the training set. If the phoneme of the word end exceeds 10 standard deviation away, we mark it as word lengthening and the silence following it as <pause> . For the filled pause, we simply relabel the silence following predefined filler words as <pause> . Finally, we exploit the disfluency detector [18] based on small vocabulary BERT to identify reparandum and interregnum (e.g. "you know", "well", "I mean") of disfluencies. The silence following the identified phrases are labeled as <pause> .

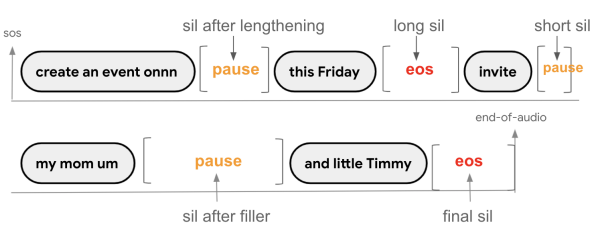

Figure 2: *An example of annotation*

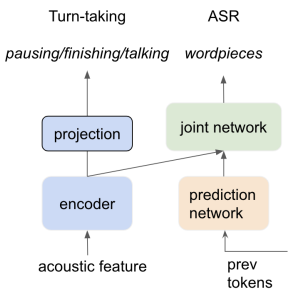

Figure 3: *Acoustic based turn-taking detector.*

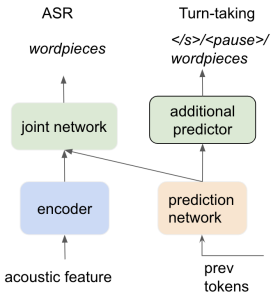

Figure 4: *Text based turn-taking predictor.*

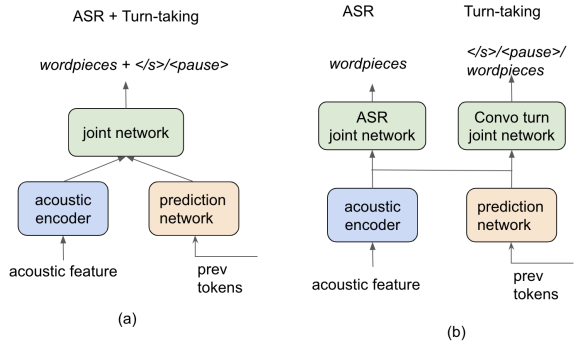

Figure 5: *(a) E2E detector (b) E2E detector with conversation joint network*

3. Models

After finishing expanding the transcript with </s> and <pause> as described in 2, we look to train the E2E model with these extra tokens to do both ASR and turn-taking in this Section. To model conversation turn-taking, we explore models that share different components of an E2E Recurrent Neural Network Transducer (RNN-T) [19] based ASR system. RNN-T consists of an encoder, a prediction network and a joint layer. The encoder consists of multiple Conformer layers [20]. The prediction network summarizes a history of previous predictions into a hidden representation. The joint layer then combines the encoder and the prediction network outputs to predict a wordpiece token given the speech inputs. In the following section, we describe different approaches to build the turn-taking detector, specifically on top of the encoder, prediction network and joint layer respectively.

3.1. Acoustic Detector

In this section, we build a frame-level turn-taking detector based on acoustic observations as illustrated in Figure 3. To build the architecture, we simply add a projection layer on top of the encoders to output the turn-taking targets, talking, pausing (<pause>) and finishing (</s>). As the encoders are shared, the architecture can achieve a better synchronization between the ASR model and the turn-taking detector, which is important to ensure correct interaction for natural conversational input.

We convert all the wordpiece symbols to the talking class and obtain the frame-level targets based on silence alignment. For each input speech frame $\mathbf{x}_t$ at timestamp t, the encoder detector computes the probability distribution of talking, pausing and finishing based on observations $\mathbf{x}_t, ...\mathbf{x}_{t-k}$ where k represents total context window received by the conformer encoder.

We can express the probability of <pause> and </s> as:

$$P(\texttt{<pause>}|\mathbf{x}_t, \cdots, \mathbf{x}_{t-k}),\ P(\texttt{</s>}|\mathbf{x}_t, \cdots, \mathbf{x}_{t-k}) \quad (1)$$

where the probabilities are thresholded to obtain the pausing or finishing decision.

3.2. Text Predictor

In this section, we investigate a turn-taking detector that is purely driven by the wordpiece sequence history while discarding the acoustic clues. Figure 4 illustrates the additional predictor consisting of a fully connected network built on top of the prediction network. As the acoustic observations are skipped, the text based detector is basically a language model predicting probability of the next word being <pause> or </s> given the sequence of word pieces already present.

To train the model, we first optimize the encoder, prediction network and joint network to predict conventional wordpiece label sequences. Next, we train the additional predictor with all the other parameters frozen. The additional predictor receives the prediction network output conditioned on N previous (non-blank) label predictions [21] and directly output the label sequences including wordpieces as well as <pause> and </s> tokens. Thus, the additional predictor is essentially a next word predictor running over the space of wordpieces plus <pause> and </s>.

During inference, we pass top hypothesis to the additional predictor to compute a probability distribution over expanded labels while only take the posterior of <pause> and </s> for turn-taking decisions, which could be expressed as:

$$P(\texttt{<pause>}|y_{u-N}, \ldots, y_u), P(\texttt{</s>}|y_{u-N}, \ldots, y_u) \quad (2)$$

3.3. E2E Detectors

Both acoustic clues and spoken words provide useful features for turn-taking decisions. To exploit both features, a straightforward approach is to build an E2E RNN-T optimizing for wordpieces, <pause> and </s> prediction as illustrated in Figure 5a, referred as E2E detector. At each time step t, the model receives a new acoustic frame x_t and outputs a probability distribution over $y_t \subset \{V \cup \texttt{<pause>} \cup \texttt{</s>}\}$, V being the wordpiece vocabulary and a blank symbol.

However, the E2E model could degrade ASR quality as the <pause> and </s> tokens do not provide informative features for wordpiece prediction, thus decrease the effective context window instead. To ensure that recognition quality is consistent with the conventional ASR, we adapt the model architecture by introducing separate joint networks for ASR joint network and conversation turn joint network as illustrated in Figure 5b, referred as E2E additional joint. The conversation turn joint network is responsible for the turn-taking decisions while the ASR joint layer decodes the wordpieces.

We perform two stages training strategy similar to text predictor in Sec. 3.2. Specifically, we first optimize the encoder, prediction network and the ASR joint layer to predict wordpiece label sequence. Next, we initialize the conversation turn joint network with the ASR joint network. The conversation turn joint network is then fine-tuned with the expanded label sequence including wordpieces, <pause> and </s> to adapt the parameters with respect to the additional loss due to extra tokens insertion. Thus, the conversation turn joint network is able to predict distributions of <pause> and </s> given the existing encoder and prediction network outputs. During inference, we rely on the ASR joint network for beam search decoding over wordpiece space:

$$y* = \arg\max_y \log P_{asr}(y|\mathbf{x}_{t-k}, \cdots, \mathbf{x}_t, y_{u-N}, \ldots, y_u) \quad (3)$$

At each time step, the conversation turn joint network computes the the probability of the <pause> and </s> given the decoding paths obtained by ASR joint network using Eq. 3. and acoustic observations. The posterior of <pause> and </s> can be expressed as:

$$\begin{aligned} &P_{convo}(\texttt{<pause>}|\mathbf{x}_{t-k}, \cdots, \mathbf{x}_t, y_{u-N}, \ldots, y_u), \\ &\quad P_{convo}(\texttt{</s>}|\mathbf{x}_{t-k}, \cdots, \mathbf{x}_t, y_{u-N}, \ldots, y_u) \end{aligned} \quad (4)$$

Unlike Eq. 1 or Eq. 2 that predicts decisions based on either the acoustic observations or the word sequences alone, Eq. 4 learns the turn-taking objectives from both. The sequence-to-sequence training is performed without alignment, thus we use the FastEmit [22] regularization to encourage paths that outputs tokens earlier.

4. Experimental Setup

4.1. Data

The training data covers utterances from short-form voice query and long-form voice transcription data. The short-form voice query data consists of around 15M utterances collected from Google's voice search product and actions on various platforms while long-form voice transcription includes 10M of utterances obtained from YouTube. The utterances are anonymized and hand-transcribed. In addition to the diverse training sets, multi-condition training (MTR) [23] are also used to further increase data diversity.

To create an evaluation set with disfluency, we first design the scripts based on common voice queries. Each script contains multiple continued queries as an example shown in Fig. 2. For each script, the speakers insert 4 different types of disfluency including random pauses, filled pauses, word lengthening and repeating phrase. Finally, the speakers manually annotate pausing or finishing labels and the corresponding timestamps. Totally, the evaluation set consists of 200 utterances recorded by 10 speakers, called natural conversation set. We also include 14K voice queries from anonymized Google's voice search set to ensure the recognition quality on a large set.

4.2. RNN-T Model Architecture

The RNN-T models use 128D log-Mel features. The encoder network architecture consists of 12 Conformer layers where each layer is of 512 dimension following [24]. The Conformer layers consist of causal convolution and left-context attention layers where 8-head attention is used in the self-attention layer and the convolution kernel size used is 15. The RNN-T decoder consists of a prediction network and a joint network with a single feed-forward layer with 640 units. The embedding prediction network [21] uses an embedding dimension of 320, and has 1.96M parameters. E2E models are trained to predict 4,096 word pieces [25] plus <pause> and </s> . We also add the FastEmit [22] regularization with a weight of 5e-3 to improve the model's prediction latency. There is no 2nd-pass used for these experiments.

4.3. Evaluation metrics

To evaluate the quality, of detectors we compute the recall and precision rate for both pausing and finishing. We first align the

hypothesized transcription and pausing/finishing to the reference to pair the predictions with true labels. Then, we could easily calculate the precision, recall and latency based on the alignment. A good recall rate of </s> is critical to ensure the queries are detected and responded. The precision rate of </s> affects if the system could interrupt user as false emission causes low precision rate. To evaluate latency, we measure the time difference between the system predicted pausing or finishing and the true labels. We only calculate the latency from the detected pausing or finishing. We also evaluate the WERs while only to ensure that adding the turn-taking detectors doesn't hurt recognition quality. The WERs of ASR without turn-taking detectors are 6.3% for the 14k voice search test set and 10.1 % for the natural conversation test set.

5. Results

In this section we present the experimental results comparing the turn-taking detectors as described in 3. We first investigate the WERs of each model on typical voice-search set and natural conversation set. Table 2 demonstrates that E2E detector could increase the WERs by 6% relatively, which implies that directly optimizing the conventional RNN-T to both tasks would hurt recognition quality as suggested in Section 3.3 while the recognition quality could be fixed by introducing additional joint network i.e. E2E add. joint in Table 2. Hence, we only compare the acoustic detector, text predictor and E2E with additional joint for the following experiments on detection accuracy and latency.

Table 2: *WER of each model. VS is the typical voice-search set; Convo is the natural conversation set.*

WER (%)	Acoustic detector	Text predictor	E2E detector	E2E add. joint
VS set	6.3	6.3	6.7	6.3
Convo set	10.1	10.1	10.6	10.1

Figure 6 reports the PR (Precision-Recall) curves for both </s> and <pause> . Upper curves are better. The curve is obtained by sweeping the thresholds of posterior of </s> and <pause> obtained by Eq. 1, 2 and 4. Figure 6 demonstrates that the system based on the E2E with additional joint is better than acoustic and text based detector, both of which degrade rapidly for the region where precision rate is over 70%. Table 3 is the optimal operating point obtained in Figure 6. Table 3 shows that although both the acoustic and text detector could achieve a good recall rate covering over 97% true </s> with a small median latency of around 100 ms, both systems have a precision rate below 70% which indicates that pausing are misclassified as finishing. The problem has been largely improved using the E2E with additional joint where precision rate has been improved by roughly 18%. The results reveal that while either acoustic observations or word sequences alone could easily identify pausing of disfluency as finishing, and combined modeling can significantly remedy the problem.

In Figure 7, we compare the 3 systems for pausing detection. Figure 7 reveals that text predictor is much worse than the other two systems on predicting pausing. This indicates that it is difficult to predict pausing based on only word sequences history. The system based on E2E with additional joint still performance best over the 3 systems where both recall and precision are over 10% better than the acoustic detector as shown in Table 4.

Table 3: *Precision, recall and latency for finishing speaking.*

model	recall (%)	precision (%)	50th latency	90th latency
Acoustic detector	97.3	67.3	90 ms	330 ms
Text predictor	97.2	66.5	90 ms	200 ms
E2E add. joint	97.5	84.7	100 ms	240 ms

Table 4: *Precision, recall and latency for pausing.*

model	recall (%)	precision (%)	50th latency	90th latency
Acoustic detector	72.5	60.0	270 ms	790 ms
Text predictor	29.8	69.7	70 ms	930 ms
E2E add. joint	84.8	77.5	300 ms	840 ms

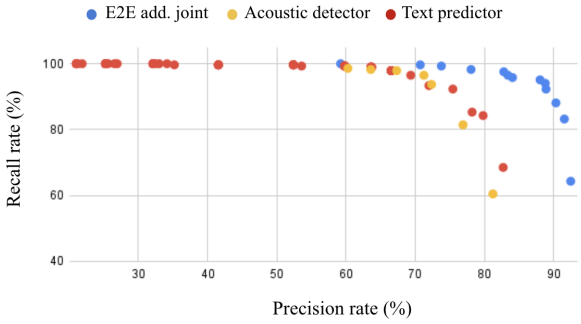

Figure 6: *Precision-Recall curve of finishing speaking.*

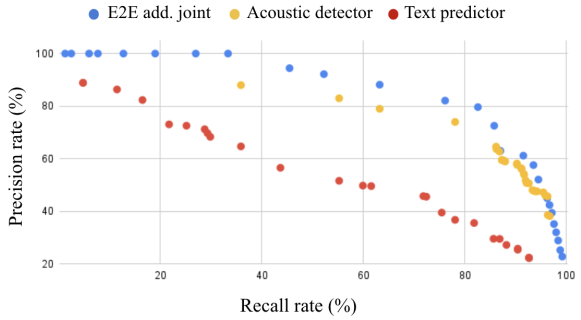

Figure 7: *Precision-Recall curve of pausing.*

6. Conclusion

In this work, we incorporate a turn-taking detector into an unified E2E RNN-Transducer by sharing the encoder and the prediction network while adapting the joint layers to optimize both the recognition and turn-taking detection. The proposed approach demonstrates over 97% recall rate and 85% precision rate on predicting true turn-taking with only 100 ms latency on difficult continued queries with 4 types of disfluencies. The analyses reveal that pure acoustic or text based predictor achieve comparable performance on detecting finishing while acoustic observations are much more useful for pausing detection.

7. Acknowledgement

We would like to thank to Jon Bloom and Jaclyn Konzelmann for natural conversation set design and Johann Rocholl and Dan Walker for providing the disfluency detector for labeling.

8. References

[1] S. Thomas, G. Saon, M. V. Segbroeck, and S. Naranyanan, "Improvements to the ibm speech activity detection system for the darpa rats program," *ICASSP*, 2015.

[2] Shuo-Yiin Chang, Rohit Prabhavalkar, Yanzhang He, Tara N Sainath, and Gabor Simko, "Joint endpointing and decoding with end-to-end models," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5626–5630.

[3] Shuo-Yiin Chang, Bo Li, and Gabor Simko, "A unified endpointer using multitask and multidomain training," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 100–106.

[4] Shuo-Yiin Chang, Bo Li, Tara N Sainath, Gabor Simko, and Carolina Parada, "Endpoint detection using grid long short-term memory networks for streaming speech recognition.," in *Interspeech*, 2017, pp. 3812–3816.

[5] Bo Li, Shuo-yiin Chang, Tara N Sainath, Ruoming Pang, Yanzhang He, Trevor Strohman, and Yonghui Wu, "Towards fast and accurate streaming end-to-end asr," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6069–6073.

[6] R. Maas, A. Rastrow, C. Ma, G. Lan, K. Goehner, G. Tiwari, S. Joseph, and B. Hoffmeister, "Combining acoustic embeddings and decoding features for end-of-utterance detection in real-time far-field speech recognition systems," *ICASSP*, 2018.

[7] D. Lala, K. Inoue, and T. Kawahara, "Evaluation of real-time deep learning turn-taking models for multiple dialogue scenarios," *ICML*, 2018.

[8] C. Liu, C. Ishi, and H. Ishiguro, "Turn-taking estimation model based on joint embedding of lexical and prosodic contents," *Interspeech*, 2017.

[9] Julian Hough Angelika Maier and and David Schlangen, "Towards deep end-of-turn prediction for situated spoken dialogue systems," *Interspeech*, 2017.

[10] R. Masumura, T. Asami, H. Masataki, R. Ishii, and R. Higashinaka, "Prediction of turn-taking using multitask learning with prediction of backchannels and fillers," *Interspeech*, 2018.

[11] D. Lala, S. Nakamura, and T. Kawahara, "Analysis of effect and timing of fillers in natural turn-taking," *Interspeech*, 2019.

[12] J. Li, Y. Wu, Y. Gaur, et al., "On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition," in *Proc. Interspeech*, 2020.

[13] Y. He, T. N. Sainath, R. Prabhavalkar, et al., "Streaming End-to-end Speech Recognition For Mobile Devices," in *Proc. ICASSP*, 2019.

[14] C.-C. Chiu, T. N. Sainath, Y. Wu, et al., "State-of-the-art Speech Recognition With Sequence-to-Sequence Models," in *Proc. ICASSP*, 2018.

[15] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proc. ASRU*, 2019.

[16] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A new training pipeline for an improved neural transducer," in *Proc. Interspeech*, 2020.

[17] G. Pundak and T. N. Sainath, "Lower frame rate neural network acoustic models," in *Proc. Interspeech*, 2016.

[18] J. C. Rocholl, V. Zayats, D. D. Walker, N. B. Murad, A. Schneider, and D. J. Liebling, "Disfluency detection with unlabeled data and small bert models," *Interspeech*, 2021.

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

[20] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[21] R. Botros and T.N. Sainath, "Tied & reduced rnn-t decoder," in *Proc. Interspeech*, 2021.

[22] Jiahui Yu, Chung-Cheng Chiu, Bo Li, Shuo-yiin Chang, Tara N Sainath, Yanzhang He, Arun Narayanan, Wei Han, Anmol Gulati, Yonghui Wu, et al., "Fastemit: Low-latency streaming asr with sequence-level emission regularization," in *Proc. ICASSP*. IEEE, 2021, pp. 6004–6008.

[23] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. N. Sainath, and M. Bacchiani, "Generated of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *Proc. Interspeech*, 2017.

[24] Arun Narayanan Tara N. Sainath, Yanzhang He et al., "An efficient streaming non-recurrent on-device end-to-end model with improvements to rare-word modeling," 2021.

[25] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in *Proc. ICASSP*, 2012.