INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Random Utterance Concatenation Based Data Augmentation for Improving Short-video Speech Recognition

Yist Y. Lin, Tao Han*, Haihua Xu*, Van Tung Pham, Yerbolat Khassanov,
Tze Yuang Chong, Yi He, Lu Lu, Zejun Ma*

ByteDance
{yist.lin0, tao.han, haihua.xu, van.pham, yerb.khass, tychong, heyi.hy, lulu0314, mazejun}@bytedance.com

Abstract

One of limitations in end-to-end automatic speech recognition (ASR) framework is its performance would be compromised if train-test utterance lengths are mismatched. In this paper, we propose an on-the-fly random utterance concatenation (RUC) based data augmentation method to alleviate train-test utterance length mismatch issue for short-video ASR task. Specifically, we are motivated by observations that our human-transcribed training utterances tend to be much shorter for short-video spontaneous speech (∼3 seconds on average), while our test utterance generated from voice activity detection front-end is much longer (∼10 seconds on average). Such a mismatch can lead to suboptimal performance. Empirically, it's observed the proposed RUC method significantly improves long utterance recognition without performance drop on short one. Overall, it achieves 5.72% word error rate reduction on average for 15 languages and improved robustness to various utterance length.

Index Terms: random utterance concatenation, data augmentation, short video, end-to-end, speech recognition

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) [1–4] framework has been predominant in both academic and industrial areas [5–7], thanks to its simplicity, compactness, as well as efficacy in modeling capacity. However, there still remains unresolved problems for E2E ASR framework. One of the problems is the learned E2E ASR models tend to overfit to the utterances that have been seen during the training. This makes the models cannot well generalize to the unseen utterances during the inference. Here, the so-called "unseen" can refer to various noise conditions [8,9], out-of-vocabulary or tail words [10,11], as well as sequence length mismatch problems [12–16], to mention a few.

To alleviate the overfitting problem, many effective solutions have been proposed. One of the earlier pioneer works is dropout [17–20], which randomly drops some of the visible or hidden nodes during training. To date, dropout is widely employed in deep learning to yield robust networks that can alleviate generalization problems. Following that, another popular technique developed to address train-test data mismatch problem is scheduled sampling [14, 21, 22], which is crucial for autoregressive attention-based encoder-decoder (AED) framework [2, 23, 24]. This is because ground-truth tokens are used as prior knowledge during training stage, while during evaluation the prior tokens might be erroneous.

Another group of works are focused on front-end data augmentation [25–29]. To let models see diverse data, speed perturbation [25], as well as artificial reverberant noise corruptions [26] are applied. Inspired by "cutout" in computer vision [30], the recently proposed SpecAugment [27], which masks a small portion of Mel coefficients along both spectral and temporal axes during training, has demonstrated significant efficacy for ASR performance.

More recently, the problem of train-test utterance length mismatch has drawn increased attention in both machine translation (MT) [31] and ASR communities [12,13,15,16]. Specifically, it is observed that the models trained with short utterances can result in significantly degraded results (a lot of deletions) when the input test utterances are much longer. To address this problem in MT, [31] proposed a multi-sentence resampling (MSR) method to concatenate sentences, yielding improved translation performance. In ASR task, a series of recipes, such as passing random state to simulate utterance concatenation [12], attention mechanism manipulation [13,23], and overlapping decoding [12, 15], have been proposed to improve long utterance recognition.

In this paper, we propose a random utterance concatenation (RUC) method to address the train-test utterance length mismatch problem in short-video speech recognition scenario[1]. We are motivated from the observation that our human transcribed utterances are too short (∼3 seconds on average), while the test utterances from the front-end voice activity detection (VAD) are much longer (∼10 seconds on average). The RUC method augments the training data by randomly concatenating two or more utterances. The intention here is to improve the robustness of ASR system by letting the E2E ASR model see more diverse utterances in length.

Our main contributions lie in the following aspects: First to the best of our knowledge, we are the first to explicitly apply on-the-fly RUC for ASR training, and reveal the effectiveness of the proposed method as a front-end data augmentation under ASR scenario. Secondly, we validate the efficacy of the proposed method on the challenging short-video ASR tasks of 15 languages, where the datasets are real, spontaneous, and noisy, and the amount of the training data is in the range from ∼1,000 to ∼32,000 hours. Besides, we have conducted extensive experiments for optimal concatenation settings, as well as a series of analysis under a wide range of train-test length mismatch conditions.

2. Relation to prior work

To alleviate the train-test utterance length mismatch problem in MT task, [31] proposed a MSR-based sentence concatenation method that is very close to the proposed RUC here. Although some analyses are conducted on Librispeech data for the ASR

*Authors equally contributed to this work.

[1] Short video refers to the one with duration of around three minutes in this paper.

10.21437/Interspeech.2023-1272

task, no actual experiment indicating the efficacy for the ASR task has been performed in [31]. Besides, compared with the MT task, utterance concatenation for the ASR task is more complicated, since the concatenated utterances are not only potentially heterogeneous in semantics, they might also be acoustically irrelevant. One cannot know for sure whether it is working without empirical study.

Train-test utterance length mismatch problem has been extensively studied in ASR community [12,13,15] recently. However, the hallmark of the work is to deal with very long-form speech to decode. For instance, the length of their target utterances can be as long as several minutes. Here, our target test utterances are shorter than 25 seconds. For such utterances, the effectiveness of their methods has not been reported. Besides, all experiments in [12, 13, 15] are conducted on RNN-T ASR framework, while our ASR models belong to the AED. Additionally, the proposed RUC is purely a data augmentation method, and it is orthogonal to the prior approaches proposed in [12, 13, 15] that mainly make efforts on model level.

3. Random utterance concatenation

The purpose of the proposed RUC method is to generate longer utterances on-the-fly by randomly concatenating utterances during training. Algorithm 1 reveals the implementation details. In Algorithm 1, the length of concatenated utterances are constrained within 300 tokens, and the maximum duration is 25 seconds in time. The "feature" in Algorithm 1 can be either raw waveform or MEL spectral coefficients. In this paper, all concatenations are performed on MEL feature level.

Algorithm 1 On-the-fly Random Utterance Concatenation

```
$S \leftarrow$ overall training steps
$B \leftarrow$ batch size of training data
$N \leftarrow$ maximum number of utterances to concatenate
for $s \leftarrow 1$ to $S$ do
    $D \leftarrow$ randomly subset buffer of overall training set
    $b \leftarrow \varnothing$
    while $|b| < B$ do
        $n \leftarrow$ random integer from 1 to $N$
        new_trans $\leftarrow \varnothing$
        new_feat $\leftarrow \varnothing$
        for $i \leftarrow 1$ to $n$ do
            (transcript, feature) $\leftarrow$ random_sample($D$)
            new_trans $\leftarrow$ new_trans.concat(transcript)
            new_feat $\leftarrow$ new_feat.concat(feature)
        end for
        b.append((new_trans, new_feat))
    end while
    do_one_step_training(b)
end for
```

The whole RUC training is divided into two stages. The first stage is a normal cross-entropy training with learning rate decay method for 200k steps. After that, we employ the RUC method to fine-tune the model with constant learning rate for another 50k steps.

4. Dataset

We employ the anonymized short-video dataset to verify the efficacy of the RUC approach for ASR training. The data is rather challenging. Not only are they spontaneous, the genres are also highly diverse. For instance, within a normal short video, majority of audios are not speech, but music, ambient noise, and other non-verbal sounds, such as laughter, coughs, breath, etc. Taking German and Swedish as examples, we observe over 90 percent of videos contain only around 3-5 short utterances that are transcribable.

Table 1 reports the overall training data statistics for 15 languages used in this work. The dataset lengths are in the range from 1k to 32k hours. The smallest training dataset is Swedish, with 1k hours. Following Swedish, Japanese and Korean are also not that large, with 1.7k and 2.5k hours respectively. While the largest is English that covers the data from different countries, including US, UK, Canada, Australia, New Zealand, and South African. Table 2 further presents the average utterance length statistics of training and test sets in terms of both time and token units for all languages. From Table 2, we can see that the average utterance length of our training data is much shorter than the test data in terms of both time and token units. Taking time unit for instance, Russian training set has the longest utterances, 5.11s on average, and with the ratio of test-train utterance duration $r \approx 2.17$. Japanese has the shortest utterances, with 2.63s on average and $r \approx 4.71$. The length mismatch here mainly attributes to two reasons. On the one hand, as was mentioned earlier, our short-video data is mostly non-speech, and

Table 1: *Training data statistics of 15 languages*

Language (ID)	Hours (K)	Utterances (M)
Burmese (my)	3.8	2.8
Dutch (nl)	4.8	5.2
Filipino (fil)	5.0	4.8
French (fr)	7.0	8.4
German (de)	9.5	10.6
Indonesian (id)	9.8	9.4
Italian (it)	9.6	7.9
Japanese (ja)	1.7	2.2
Korean (ko)	2.5	2.2
Polish (pl)	9.6	9.5
Portuguese (pt)	10.0	13.2
Russian (ru)	9.0	6.2
Swedish (sv)	1.0	1.1
Vietnamese (vi)	9.7	11.7
English (en)	32.7	24.4

Table 2: *Average utterance length statistics (mean ± standard deviation) in training and test sets for 15 languages*

ID	Training		Test	
	Duration (s)	#Tokens	Duration (s)	#Tokens
my	4.62 ± 3.49	13.47 ± 11.87	10.67 ± 4.51	29.2 ± 18.9
nl	3.19 ± 2.63	9.12 ± 7.76	10.93 ± 4.72	24.2 ± 17.1
fil	3.51 ± 2.82	9.88 ± 8.46	9.98 ± 4.26	21.3 ± 13.3
fr	2.82 ± 2.44	11.30 ± 10.08	10.47 ± 4.64	34.1 ± 23.1
de	3.08 ± 2.86	9.74 ± 9.44	10.22 ± 4.21	26.7 ± 15.9
id	3.57 ± 3.15	9.09 ± 8.23	11.35 ± 4.54	20.8 ± 13.9
it	4.14 ± 3.22	11.53 ± 10.11	10.85 ± 4.71	26.6 ± 17.5
ja	2.63 ± 2.56	14.95 ± 14.58	12.39 ± 4.11	78.0 ± 39.6
ko	3.89 ± 3.28	8.95 ± 8.59	10.33 ± 4.80	17.7 ± 13.9
pl	3.50 ± 2.88	11.81 ± 10.39	10.41 ± 4.61	32.7 ± 20.9
pt	2.73 ± 2.08	8.76 ± 7.09	10.07 ± 3.97	29.8 ± 15.7
ru	5.11 ± 3.99	14.45 ± 12.52	11.11 ± 4.54	23.9 ± 15.4
sv	3.35 ± 2.70	10.30 ± 9.06	10.59 ± 4.58	29.0 ± 19.4
vi	2.88 ± 2.58	12.47 ± 11.28	10.70 ± 4.73	35.8 ± 25.5
en	4.84 ± 4.23	15.64 ± 15.29	11.74 ± 3.73	39.8 ± 17.9

human transcribers tend to have short utterances to transcribe for training data preparation. On the other hand, it is related with our specific test settings. Our incoming test data is an entire short video, and we rely on a NN-based VAD front-end to first remove non-speech parts, then split longer speech segments into smaller ones. Finally, they are fed to the ASR engine. To avoid intra-speech segmentation or speech missing, the VAD tends to output longer speech to the ASR engine, which results in train-test length mismatch.

5. Experiments and results

5.1. Modeling

The E2E ASR models employed in this work are based on attention-based encoder-decoder architecture, which are similar to the ones used in [32]. Specifically, the encoder is Transformer, while the decoder is LSTM. Transformer's {layer, dim, head} parameters are {18, 512, 8}, and the feed-forward network dimension is set to 2048 with the GLU activations. The LSTM decoder has four layers with 1024 cells per layer. For robust training, we employed both variational noise (VN) [33] and SpecAugment [34] methods. The VN training is activated after 10k steps. The SpecAugment is activated after 2k steps, and its frequency {F, m_F} and temporal {W, m_T, p} parameters are set to {27, 2} and {100, 1, 0.1} respectively. During inference, the beam size is fixed to 10, and the best length normalization factor [35] is selected for each language independently in [0.0, 0.8] range. All ASR models employ word piece model [36, 37] with the vocabulary size being 3-7k.

5.2. Results

Table 3 reports the WER and corresponding WER reduction (WERR) results (for N={1, 4, 6, 8}) using the proposed RUC method on 15 languages. In table 3, we report 4 categories of experiments, where N={4, 6, 8} means we randomly concatenate up to 4, 6, 8 utterances for the proposed method. As a contrast, N=1 refers to a continual training without any utterance concatenation. As shown in Table 3, continual training without the proposed method gets only 0.62% WERR on average, while the proposed RUC data augmentation method makes consistent significant WERR improvement with three concatenation settings. The general trend is shown to be the larger N the bigger WERR. Specifically the WERR is from 4.24% up to 5.72% on average. In Table 3, the RUC has not achieved similar WERRs for each language. The best WERRs are obtained on Vietnamese (vi), Portuguese (pt), German (de), and Burmese (my) 4 languages, and their best WERRs are over 10%. Interestingly, it gains 22.53% WERR on Vietnamese. We hypothesize such a huge WERR could be related with too short utterances ($\sim 2.88s$) in the training dataset. On the opposite side, there are almost no WERRs for English (en), Russian (ru), Polish (pl), and Dutch (nl) 4 languages, and the average utterance length of these 4 languages are over 3.0s. These might partially mean RUC is potential useful for the training dataset that has a lot of shorter utterances.

Figure 1 plots the WERR versus train-test utterance duration ratio for each of 15 languages. Denoting R as train-test average utterance duration ratio, we can observe from Figure 1 that when $R \in [0.5, 1.0]$, all language ASR models obtain increased WERRs except for Korean, and Polish ASR models. Particularly, the latter shows clear performance drop trend. Interestingly, as $R \in [1.0, 2.0]$, there is no obvious WERR deterioration for any language, and on the contrary, the best performing languages in Table 3 still get further WERR. The result in Figure 1 indicates employing RUC to produce training utterances as one to two times long as test utterances (R lies in [1.0, 2.0]) could yield improved ASR performance.

Table 3: *WER (%) and corresponding WERR (%) for 15 languages using the proposed RUC data augmentation method, where N=1 means 50k steps of continual training, and no utterance concatenation being performed*

ID	Baseline (WER)	N=1	N=4	N=6	N=8
my	20.65	-4.00	5.94	8.36	10.15
nl	23.90	-2.30	2.73	1.94	1.10
fil	26.27	1.54	2.74	2.17	2.98
fr	19.35	1.29	5.13	6.08	6.87
de	15.05	4.25	7.73	9.59	10.12
id	21.79	2.94	5.03	6.06	5.58
it	18.27	2.38	4.18	3.85	3.56
ja	17.66	-2.89	-1.45	3.57	4.10
ko	18.70	2.73	1.52	4.10	5.42
pl	13.12	3.16	0.08	-1.42	1.30
pt	12.72	1.77	9.07	8.94	10.06
ru	19.22	-3.59	-0.40	1.37	0.68
sv	24.88	2.97	5.08	1.80	1.65
vi	26.73	1.48	17.23	22.38	22.53
en	9.62	-2.39	-1.04	-0.49	-0.31
Avg.	19.20	0.62	4.24	5.22	5.72

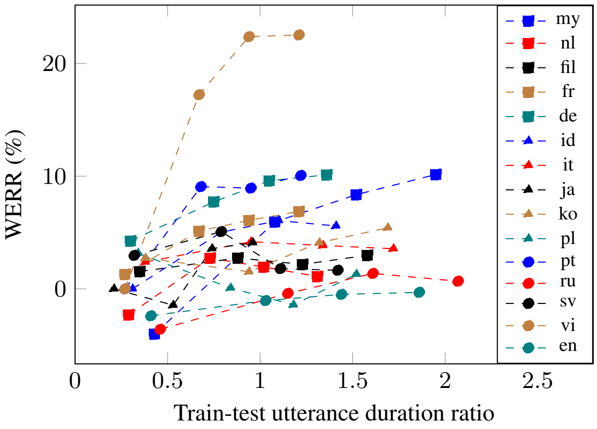


Figure 1: *WERR vs. train-test utterance duration ratio (R) for all languages*

6. Analysis

Train-test mismatch is unavoidable in real applications. To simulate such a mismatch, we have attempted different test utterance length by VAD, namely, 15s, 12s, 10s, 7s[2], so as to examine the robustness of the proposed method. We find after RUC training the performance fluctuation is considerably reduced. Table 4 reveals the regarding efficacy via a contrast between ASR models with RUC and without RUC training on 4 languages. We can observe from Table 4 not only is WER performance improved but the WER standard deviations (SDs) with different VAD settings are also remarkably reduced respectively. We note that the WER SDs of the Korean (ko) are

[2] Ns denotes the average utterance length produced with a VAD setting is N seconds.

Table 4: *WER (%) under different utterance length by VAD settings and their standard deviations (SD) before and after the proposed RUC training for 4 languages*

ID	Without RUC					With RUC				
	15s	12s	10s	7s	SD	15s	12s	10s	7s	SD
pt	16.61	16.54	15.72	15.37	**0.61**	16.02	16.12	15.72	15.59	**0.25**
vi	19.76	20.24	17.21	15.12	**2.38**	15.47	15.61	13.62	13.68	**1.09**
ja	15.04	15.11	13.93	13.40	**0.84**	13.59	13.57	12.96	13.66	**0.33**
ko	12.19	12.27	11.93	12.31	**0.17**	11.93	11.77	11.82	12.24	**0.21**

changed little since its WERs between different VAD settings are already rather small, and it gets minor increased after RUC training, from 0.17 to 0.21.

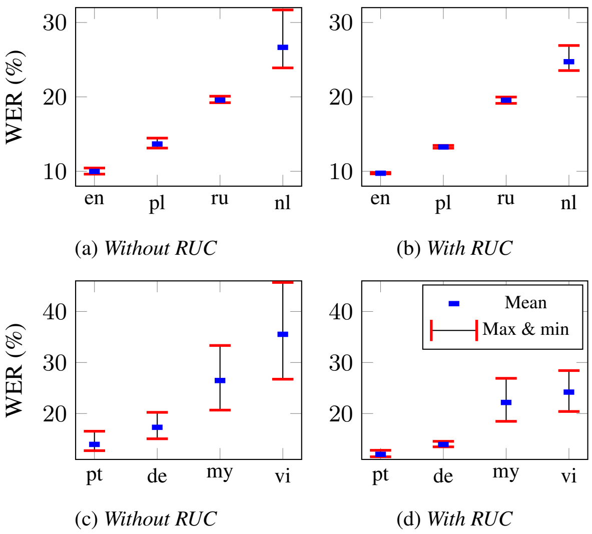
One of the bad effects of train-test utterance length mismatch, particularly training utterance being much shorter than test ones, is that many more deletion errors will arise during inference [31]. To alleviate deletion errors, one would simply employ length normalization therapy which basically awards longer utterances. We use the length normalization method as advocated in [35] in this work, and it is formulated as follows:

$$s(\boldsymbol{y}, \boldsymbol{x}) = \sum_{i=1}^{m} \log P(y_i \mid y_{1:i}, \boldsymbol{x}) \qquad (1)$$

$$s'(\boldsymbol{y}, \boldsymbol{x}) = s(\boldsymbol{y}, \boldsymbol{x}) \; / \; \frac{(5 + |\boldsymbol{y}|)^{\alpha}}{(5+1)^{\alpha}} \qquad (2)$$

where $|\boldsymbol{y}|$ refers to the token length of the decoded hypothesis, and α is the so-called length normalization hyper-parameter that controls the dynamic range of the denominator, normally in the range of [0.0, 0.8] in our work. Eq. 1 is the normal AED decoding formula, while Eq. 2 is the implementation formula of Eq. 1 in practice. We found that after RUC training, the ASR model is less reliant on normalization, that is, we found the WER gaps between different normalization factors are much smaller after RUC training. Figure 2 illustrates the WER dynamic range on two groups of languages. One group of languages that obtains the least WERR from RUC training are English, Polish, Russian, and Dutch (Figure 2a and 2b), while the other group that has achieved best WERR are Portuguese, German, Burmese, and Vietnamese (Figure 2c and 2d) respectively. From Figure 2, we notice that the WER change dynamics are obviously smaller after RUC training compared with the case of which no RUC training is performed on either group of languages. This indicates RUC training brings more robust ASR models, which would be beneficial when our ASR engine receives diverse incoming speech.

Table 3 has clearly revealed the effectiveness of the proposed RUC training on overall WERR, and now we are curious about if such a performance improvement occurs on the overall test utterances with diverse utterance length distribution. To take a closer look at what has happened with more details, we perform RUC on a predefined Swedish `dev` set, and compare the results of different ASR models with or without RUC training. Figure 3 plots the details of WER versus utterance length on Swedish. From Figure 3, it is interesting to note that the RUC training method can yield comparable WER results with the normal training method on shorter test utterances whose length is below 20 tokens. As test utterances are getting longer and longer, the gap between the proposed and conventional methods appears and tends to enlarge. More specifically, the WER gaps begin to appear for utterances in 20-40 token area, and the proposed RUC method prevails. As the utterance length is in 40-60 token area, the gap enlarges obviously. Finally, when test utterance length is over 60 tokens, the normal ASR models without RUC method yields deteriorating WER results, while the RUC method consistently gets improved WERR. This further demonstrates the effectiveness of the proposed RUC data augmentation method since it is beneficial to the overall test set with wide distribution of utterance length.



(a) *Without RUC*

(b) *With RUC*

(c) *Without RUC*

(d) *With RUC*

Figure 2: *Illustration of RUC efficacy on length normalization for inference through contrasts between those with and without RUC training.*

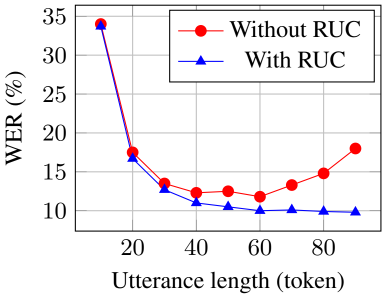


Figure 3: *Illustration of WER versus test utterance length with and without the proposed RUC method on Swedish.*

7. Conclusion

In this work, we proposed an on-the-fly random utterance concatenation method as front-end data augmentation for improving short-video speech recognition. Specifically, the proposed method addresses train-test utterance length mismatch originated from the situation in which incoming test utterance length is much longer than that in the training set. We demonstrated its efficacy using diverse ASR models from 15 languages whose datasets are in the range of 1k to 32k hours. With the method, we achieved up to 5.72% WER reduction on average for the overall languages. By further analysis, we found the proposed data augmentation method can make the ASR model less sensitive to length normalization, which potentially proves that the ASR models are more robust to diverse environments. Moreover, the proposed method is beneficial to long utterance decoding without any performance drop on short utterance.

8. References

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

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP 2016*, 2016, pp. 4960–4964.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of the 31st NIPS*, 2017, pp. 6000–6010.

[4] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-3015

[5] Y. He, T. N. Sainath, R. Prabhavalkar *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*. IEEE, 2019.

[6] T. N. Sainath, Y. He, B. Li *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. of ICASSP*. IEEE, 2020.

[7] B. Li, T. Sainath, R. Pang, S.-Y. Chang, Q. Xu, T. Strohman, V. Chen, Q. Liang, H. Liu, Y. He, P. Haghani, and S. Bidichandani, "A Language Agnostic Multilingual Streaming On-Device ASR System," in *Proc. Interspeech 2022*, 2022, pp. 3188–3192.

[8] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*. IEEE, 2017.

[9] S. Sun, P. Guo, L. Xie, and M.-Y. Hwang, "Adversarial regularization for attention based end-to-end robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1826–1838, 2019.

[10] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *ICASSP 2021*, 2021, pp. 5674–5678.

[11] C. Peyser, S. Mavandadi, T. N. Sainath, J. Apfel, R. Pang, and S. Kumar, "Improving Tail Performance of a Deliberation E2E ASR Model Using a Large Text Corpus," in *Proc. Interspeech 2020*, 2020, pp. 4921–4925. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1465

[12] A. Narayanan, R. Prabhavalkar, C.-C. Chiu, D. Rybach, T. N. Sainath, and T. Strohman, "Recognizing long-form speech using streaming end-to-end models," in *ASRU 2019*, 2019, pp. 920–927.

[13] C.-C. Chiu, W. Han *et al.*, "A comparison of end-to-end models for long-form speech recognition," in *ASRU*. IEEE, 2019.

[14] P. Zhou, R. Fan, W. Chen, and J. Jia, "Improving generalization of transformer for speech recognition with parallel schedule sampling and relative positional embedding," *arXiv:1911.00203*, 2019.

[15] C.-C. Chiu, A. Narayanan, W. Han, R. Prabhavalkar, Y. Zhang, N. Jaitly, R. Pang, T. N. Sainath, P. Nguyen, L. Cao, and Y. Wu, "Rnn-t models fail to generalize to out-of-domain audio: Causes and solutions," in *SLT 2021*, 2021, pp. 873–880.

[16] Z. Lu, Y. Pan *et al.*, "Input length matters: Improving rnn-t and mwer training for long-form telephony speech recognition," *arXiv:2110.03841*, 2021.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *Proc. of ICASSP*. IEEE, 2013.

[19] T. Moon, H. Choi, H. Lee, and I. Song, "Rnndrop: A novel dropout for rnns in asr," in *Proc. of ASRU*. IEEE, 2015, pp. 65–70.

[20] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with lstms." in *INTERSPEECH*, 2017.

[21] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Advances in NIPS*, vol. 28, 2015.

[22] C.-C. Chiu, T. N. Sainath, Y. Wu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. of ICASSP*. IEEE, 2018.

[23] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in NIPS*, vol. 28, 2015.

[24] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *in Proc. of ICASSP*. IEEE, 2016.

[25] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH*, 2015.

[26] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*. IEEE, 2017.

[27] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. of INTERSPEECH*, 2019.

[28] A. Rosenberg, Y. Zhang *et al.*, "Speech recognition with augmented synthesized speech," in *Proc. of ASRU*. IEEE, 2019.

[29] T.-S. Nguyen, S. Stueker, J. Niehues, and A. Waibel, "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation," in *Proc. of ICASSP*. IEEE, 2020.

[30] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv:1708.04552*, 2017.

[31] I. Provilkov and A. Malinin, "Multi-Sentence Resampling: A Simple Approach to Alleviate Dataset Length Bias and Beam-Search Degradation," in *Proc. EMNLP 2021*, 2021, pp. 8612–8621. [Online]. Available: https://aclanthology.org/2021.emnlp-main.677

[32] Y. Liu, R. Ma *et al.*, "Internal language model estimation through explicit context vector learning for attention-based encoder-decoder asr," in *Proc. INTERSPEECH*, 2022.

[33] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 Proc. of ICASSP*. IEEE, 2013.

[34] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[35] Y. Wu, M. Schuster *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv:1609.08144*, 2016.

[36] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th ACL*, 2016.

[37] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP 2018: System Demonstrations*, 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012