Improving Deliberation by Text-Only and Semi-Supervised Training

Ke Hu, Tara N. Sainath, Yanzhang He, Rohit Prabhavalkar, Trevor Strohman, Sepand Mavandadi, Weiran Wang

Google LLC, USA

huk@google.com

Abstract

Text-only and semi-supervised training based on audio-only data has gained popularity recently due to the wide availability of unlabeled text and speech data. In this work, we propose incorporating text-only and semi-supervised training into an attention-based deliberation model. By incorporating text-only data in training a bidirectional encoder representation from transformer (BERT) for the deliberation text encoder, and large-scale text-to-speech and audio-only utterances using joint acoustic and text decoder (JATD) and semi-supervised training, we achieved 4%-12% WER reduction for various tasks compared to the baseline deliberation. Compared to a state-of-the-art language model (LM) rescoring method, the deliberation model reduces the Google Voice Search WER by 11% relative. We show that the deliberation model also achieves a positive human side-by-side evaluation compared to the state-of-the-art LM rescorer with reasonable endpointer latencies.

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) models have made tremendous improvements in recent years [1, 2, 3, 4, 5, 6, 7, 8]. In a state-of-the-art system [1], a neural language model (LM) is used to rescore a cascaded encoder model and outperforms a conventional ASR system in both Google Voice Search (VS) and rare word recognition quality, as well as latency. The LM in [1] is trained using billions of text-only data and proves to improve rare word recognition quality. While LM relies on only text hypotheses for rescoring, deliberation models have been recently proposed for second-pass rescoring using both text hypotheses and audio [9, 10]. Compared to LM training, there has been few attempts at incorporating widely available text-only or audio-only data in deliberation (see [11]). In this work, we research various ways to utilize large-scale text-only and semi-supervised data for deliberation training.

While the LM in [1] uses causal conformer layers, bidirectional textual context is incorporated by using bidirectional encoder representations from transformers (BERT) [12, 13]. In addition to LM rescoring, neural correction models train text-to-text models to predict targets based on estimated transcripts. For example, a BERT model is used in [14] to initialize a transformer neural correction model. To increase diversity, text-to-speech (TTS) utterances are decoded to generate text hypotheses to train a transformer correction model [4]. LSTM models are used similarly in [15]. However, since neural correction only relies on text, its correction capability is potentially limited and thus only used for spelling correction.

Instead of training external modules such as LMs, several recent studies incorporate text-only data into supervised training to jointly train E2E models [16, 11, 17, 18, 19, 20]. For example, text-only data has been used to train speech encoders [16, 18]. [16] leverages text-only data represented as phonemes and masked by noise, and uses them as inputs to predict the corresponding text using a shared encoder with ASR. In [18], either text-only or speech-only data have been used in a speech-text joint training to pre-train an encoder for a downstream task such as ASR. On the other hand, ASR decoders have also been modified for text-only training. [11] extends a joint acoustic and text decoder (JATD) from the Listen, Attend and Spell (LAS) [21] to a deliberation decoder, and uses text-only data (or synthesized utterances) to train the decoder with fixed context vectors. [17] modifies the transformer decoder to have only self-attention (except for the last layer) so they can be trained by text-only data.

Besides text-only data, audio-only data is also widely available and thus used to assist ASR training [22, 23, 24, 25, 26]. For example, by pre-training speech encoders using audio-only data, wav2vec [22, 23] achieves competitive results by using a small amount of labeled data. In [24], the authors use large-size models up to billions of parameters in semi-supervised learning using unlabeled data combined with a small portion of labeled data. The idea of noisy student training is explored in [25, 26], where a bidirectional teacher generates training data for a streaming student by using noisy corrupted inputs.

In this work, we propose to incorporate text-only data to pre-training the deliberation text encoder in a masked language model (MLM) task similar to BERT [27]. Our results show that pretraining a conformer text encoder with large enough size significantly improves recognition for both Voice Search and long-tail words. In addition to the text encoder, we also synthesize large-scale text-only data (84M) to TTS utterances in training the deliberation decoder using JATD [11]. Third, since deliberation attends to encoded audio, we perform large-scale semi-supervised training using 500M unlabeled speech utterances from Google Voice Search domain and transcribed using a conventional model. With all the proposed techniques, we achieved 4%-12% WER reductions for various test sets compared to the deliberation baseline. Compared to a state-of-the-art LM rescorer [1], our deliberation model performs 11% relative better in VS, 16% for the SxS test set, and competitively for long-tail. A human side-by-side comparison shows the deliberation performs significantly better than LM rescoring with reasonable endpointing latencies.

2. Modeling Improvement

2.1. Model Overview

Our model is illustrated in Fig. 1. Note that different from [9, 10], the deliberation decoder is based on the non-causal encoder [2] instead of a causal encoder. The decoder attends to both the non-causal encoder output ($\mathbf{e}$) and hypotheses ($\mathbf{y_r}$) from the non-causal path, i.e., decoded using non-causal encoder. The non-causal encoder often has a right-context for better recognition quality [2]. We use a conformer encoder [28] as

the text encoder. A two-source attention LAS decoder is used as the deliberation decoder, similar to [9]. The decoder can be used for either re-decoding or rescoring. The deliberation model in this work does not stream compared to [29], as we focus on text-only and semi-supervised training.

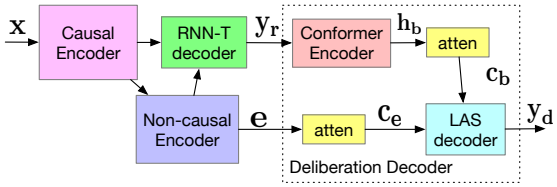

Figure 1: *Deliberation based on cascaded encoders.*

2.2. BERT Training Based on Text-Only Data

2.2.1. Pretrained BERT Text Encoder

The conformer encoder in deliberation in Fig. 1 is a text encoder which takes text hypotheses ($\mathbf{y_r}$) as inputs and outputs text encodings ($\mathbf{h_b}$). In regular deliberation training [9, 10], we randomly initialize the parameters of the text encoder and train it jointly with other parts of the deliberation model. The training uses only supervised data. In this work, to leverage large amounts of text-only data, we propose to pretrain the text encoder alone by masking the text-only inputs and then predicting the masked tokens using a cross entropy (CE) loss, similar to BERT [27].

Given an input text sequence, we first tokenize it into word-pieces, and randomly choose 15% of tokens, similar to [27]. Each token is then replaced with `[MASK]` for 80% of the time, a random token for 10% of the time, or kept unchanged for 10% of the time. Since our input hypotheses are single sentences, we use only one segment for each BERT input. We use `<sos>` to replace the special classification token `[CLS]`, and use `<eos>` for `[SEP]`. The `<sos>` and `<eos>` share the same vocabulary as the cascaded encoder model to match BERT inputs to the format of first-pass hypotheses. Each input sequence is padded to a fixed length, and padded tokens are not used for computing the CE loss. We reuse the token id of `<epsilon>` for `[MASK]` since it does not appear in our hypotheses which contain only non-blank labels. Lastly, the task of next sentence prediction is removed from our task since we only have single text sentences. Instead of transformer layers, we use conformer layers [28] for the text encoder. We refer to a text encoder trained in this way as a pretrained BERT.

The pretrained BERT is then used as the deliberation text encoder, and we update the parameters of the BERT and deliberation decoder jointly in training. The cascaded encoder is kept frozen. We have also tried freezing the entire or part of the pretrained BERT (e.g. layers close to inputs) and update the rest of the deliberation decoder, but this leads to worse quality.

2.2.2. Masking First-Pass Hypotheses

Similar to the MLM idea in BERT [27], we also try to increase the diversity of the text encoder inputs by randomly masking tokens in the input hypotheses. This aims to increase the diversity of the data seen by the text encoder. Note that this is different from the pretraining in Sect. 2.2.1 because here we do not use any external data to pre-train the text encoder. We predict the text targets use the CE loss in training, and do not use any masking in inference. We show in Sect. 3.2.5 that simply masking a small portion of first-pass hypothesis tokens in training improves over the baseline deliberation in certain conditions.

2.3. Large-Scale TTS and Semi-Supervised Training

2.3.1. Large Scale TTS Training

Text-only data can also be converted to TTS utterances for training the whole deliberation decoder. We employ JATD training [11] and scale it up using text data sampled from multiple domains, i.e., 51M, 20M, 1.6M, 0.6M, and 11M text sentences from Maps, News, Play, Search and YouTube domains, respectively. In comparison, [11] uses only 4.6M samples from the Maps domain. The text sentences are then converted to speech by using a multi-speaker TTS system [30]. In our large scale training, we mix the TTS data and supervised training data in a 1:9 ratio. The deliberation decoder is trained from scratch using the mixed data. We find that sampling data using probabilities proportional to their amount gives the best result. To differentiate supervised and TTS data, we use only data from a single domain for each training batch. When the data is supervised, we compute both audio and text attention during training, and otherwise use fixed context vectors to replace both audio and text attention for TTS data. We show that Maps-only data improves the corresponding domain quality, but degrade other domains, resulting similar average performances. However, by using data from all domains, JATD significantly improves all long-tail sets on average as well as the Maps domain (details in Sect. 3.2.4).

2.3.2. Large Scale Semi-Supervised Training

Apart from text-only data, we also explore using large-scale unlabeled audio utterances to improve training, inspired by [24, 25]. In total, we have 500M audio-only utterances sampled from the Search domain and then generate estimated transcripts using a state-of-the-art hybrid model [31] for training. The model employs a state-of-the-art language model for decoding and tends to generates quality training data for long-tail words. The utterances are not augmented by any noise. Note that in later experiments (Sect. 4.1), we also mix the large-scale TTS data (84M) and the semi-supervised data (500M) in deliberation training. As far as we know this is the first attempt to experiment with large-scale TTS and semi-supervised training for deliberation. In training, we use the TTS utterances as 10% of all data, semi-supervised data as 10%, and train the deliberation model from scratch.

3. Experiments

We perform our experiments using large-scale data [32] based on a state-of-the-art cascaded encoder model [33].

3.1. Modeling Details

3.1.1. Baseline Deliberation Model

Our deliberation model is based on a cascaded encoder baseline [33] which consists of 17 causal conformer layers and 5 non-causal layers. Each causal layer has a model dimension of 512 with 8-headed self-attention. The five non-causal layers has a total of right context of 0.9s. We use an embedding prediction network as in [34]. The cascaded encoder model is trained to predict 4,096 lowercase wordpieces [35].

Our LAS-based deliberation decoder attends to non-causal encoder outputs and hypotheses decoded using the non-causal encoder. For efficiency, we use 4 first-pass hypotheses. The text encoder is a 2-layer 640-D conformer encoder with a two-token right-context, totaling around 12M parameters. We use 8-headed attention for both audio and hypotheses. The deliberation decoder consists of 2 LSTM layers (similar to [9]), where

each layer has 2,048 hidden units followed by 640-dimensional projection. A 4,096-dimensional softmax is then used to predict the same wordpieces as the baseline cascaded encoder. The decoder has around 42M parameters.

An input speech waveform is divided into 32-ms segments using hanning windows at a rate of 10 ms to compute 128-D log-Mel features. Each log-Mel feature is then stacked with three previous frames to form a 512-D vector, which is then downsampled to a 30-ms frame rate as input features.

3.1.2. Training Data

Our multi-domain (MD) supervised training data consists of around 300M utterances described in [32]. The utterances are sampled from multiple domains, and are anonymized and hand-transcribed except for YouTube where utterances are generated using a semi-supervised method [36]. In total, we have ~400k hours of training data. We also increase the data diversity by using multi-condition training [37] such that the utterance signal-to-noise ratio (SNR) is between 0dB and 30dB. We also use mixed-bandwidth utterances at 8kHz or 16 kHz [38], and SpecAug [39]. For text-only training, we use a text corpus which contains more than 100B sentences [1] to train the BERT text encoder described in Sect. 2.2.1. The text-only data spans multiple domains including Maps, News, Play, Search and YouTube.

3.1.3. Test Data

We use three sets for evaluation. The Voice Search (VS) test set contains ~14K anonymized and hand-transcribed utterances sampled from general Google Voice Search traffic. A SxS test set contains around 900 utterances where an E2E model [40] performs inferior to a state-of-the-art hybrid model [31]. To focus on long-tail word recognition, we use a long-tail (LT) test set described in [41]. The LT utterances are synthesized using text sentences containing words rare in multi-domain training, or with surprising pronunciations [41]. The utterances range across multiple domains such as Maps, News, Play, Search and YouTube, totaling 200K. In the following ablation studies, we use a subset of LT, called Rare Proper Noun Maps (RPNM), to represent the LT set for experiment efficiency. The performance of RPNM usually correlates well with the full LT set.

3.2. Ablation Studies

3.2.1. Pretrained BERT Text Encoder

We compare three pretrained BERT (PTB) text encoders, described in Sect. 2.2.1, with large, medium and small sizes. The PTBs have 12, 4, and 2 conformer layers with a model dimension of 512, corresponding to 76M, 32M, and 12M parameters, respectively. The conformer right context is set to 30 tokens to become "bidirectional". We use the same wordpiece model as the deliberation for tokenization. We can see in Table 1 that the large PTB performs the best, with a VS WER of 4.6%. The WER improvement is uniform for all test sets, ranging from 4% to 12% compared to the baseline deliberation. We note that when the size of the PTB reduces, the improvement reduces gradually. When using a small PTB with the same size as the baseline deliberation, we did not see any benefits from pretraining. This indicates that a relatively large BERT may be needed to leverage the large amount of unpaired text data.

To further analyze whether the improvement is due to increased size of the BERT, we remove pretraining (PT) for all PTBs. We see in Table 1 that there is significant regression in large and medium PTB scenarios for all test sets. In the large BERT scenario, we also note that the model becomes hard to train without text-only pretraining, resulting worse performance than the medium-size BERT. In addition, the small non-pretrained BERT performs similar to the baseline, indicating our small conformer BERT has a similar performance as the original conformer encoder. In addition, we have also tried even larger BERT with 16 and 24 layers but none of them obtained uniform improvements for all test sets compared to the 12L BERT. Considering computation efficiency, we choose the 12L BERT as the text encoder for the following experiments.

Table 1: *WERs (%) of deliberation using pretrained and non-pretrained BERT text encoders of different sizes.*

Model	# Text Enc. Layers	WER (%)		
		VS	SxS	RPNM
Deliberation	2L	4.8	26.9	12.0
+ Large PTB	12L	**4.6**	**23.6**	**11.1**
- PT		5.3	29	13.2
+ Medium PTB	4L	4.7	25.4	11.5
- PT		4.8	26.5	12.0
+ Small PTB	2L	4.8	26.3	11.9
- PT		4.8	26.3	11.9

3.2.2. Semi-Supervised Training

As described in Sect. 2.3.2, we use 500M semi-supervised utterances with transcripts generated using a hybrid model [31] for training. We mix the semi-supervised data with the supervised data using a 1:9 ratio and train the deliberation model from scratch. The semi-supervised data mainly improves VS WER by 4% relatively (4.6% $\rightarrow$ 4.4% in Table 2). This is probably because our unlabeled data is from the Voice Search domain. In this study, our semi-supervised data size is relatively small compared to text-only data, in future we plan to generate more semi-supervised data using more powerful teachers.

Table 2: *WERs (%) by semi-supervised training.*

Model	# Text Enc. Layers	WER (%)		
		VS	SxS	RPNM
Large PTB	12L	4.6	23.6	11.1
+ Semi-sup. data		**4.4**	**23.6**	**10.9**

3.2.3. Rescoring

So far, our decoding is done by beam search. To compare to LM rescoring later (Sect. 4.2), we use deliberation to rescore the first-pass hypotheses in a teacher-forcing fashion [40]. Compared to Table 2, the rescoring WERs are 4.6%, 25.5%, and 10.9%, for VS, SxS, and RPNM, respectively. This is expected according to our previous findings [9, 10]. We use the deliberation rescorer for later experiments.

3.2.4. JATD

In Table 3, we see that by using only Maps data (JATD-Maps), JATD improve significantly on the RPNM test set, which is a set focusing on Maps. But there is no improvement in LT, indicating potential regression for other domains. We thus perform large-scale JATD training for all domains described in Sec. 3.1.2 (JATD-All). Table 3 shows that we have achieved significant improvements on both RPNM and LT. The improvement is around 9% relative for the LT set. Note that our deliberation rescoring baseline here does not incorporate pretrained BERT. Similar to [11], we notice VS and SxS results do not change significantly.

Table 3: *WERs (%) by rescoring using JATD training.*

Model	# Text Enc. Layers	WER (%)			
		VS	SxS	RPNM	LT
Delib. Rescoring	2L	4.9	27.6	12.0	30.7
+ JATD-Maps		5.0	26.8	10.4	30.7
+ JATD-All		5.0	26.9	**10.3**	**27.8**

3.2.5. Apply Masking to Hypotheses

To increase the diversity of data for text encoder training, we have also tried applying masking to first-pass hypotheses, similar to the MLM idea in [27]. Specifically, we mask around 2% of hypothesis tokens, randomize only 0.01% tokens, and leave the rest unchanged. We have tried other ratios but did not find improvement. We experiment masking to three deliberation models in Table 4. Overall, we notice that masking only improves the vanilla deliberation rescoring by 3.6% relative for the SxS set. When other techniques such as pretrained BERT or JATD are used, masking degrades the SxS and LT performance. We thus will not include this in our final system but recommend as a convenient approach to increase data diversity for the baseline deliberation.

Table 4: *WERs (%) using masking for first-pass hypotheses.*

Model	# Text Enc. Layers	WER (%)		
		VS	SxS	LT
Delib. Rescoring	2L	4.9	27.6	30.7
+ MLM		5.0	**26.6**	30.6
Delib. + PTB	12L	4.8	24.9	29.5
+ MLM		4.8	25.8	29.7
Delib. + JATD-All	2L	5.0	26.9	27.8
+ MLM		5.0	28.0	27.9

4. Comparison

4.1. WER Comparisons

In Table 5, we compare the cascaded encoder baseline (`B0`) to the baseline deliberation model (`B1`) and deliberation with proposed training techniques (`E1`-`E3`). The cascaded encoder model (`B0`) is exactly the first-pass model used for deliberation.

First, we see that our best-performing deliberation model (`E3`), with all techniques proposed in the paper, performs significantly better than the deliberation baseline (`B1`), reducing WERs by 4.1%, 6.5%, and 11.7%, for VS, SxS, and LT test sets, respectively. The improvement is more prominent for long-tail sets, indicating the text-only and semi-supervised training is especially effective for long-tail. Compared to cascaded encoder (`B0`), our WER improvement is up to 15%. For individual techniques, we see that JATD and semi-supervised training improves LT significantly by around 10% relative. BERT pretraining improves VS significantly (6% relative), and the lack of LT improvement is probably because JATD already does well in long-tail.

In Table 5, we also compare to a LM rescoring model (`B2`) similar to [1], which consists of 12 conformer layers. The LM has a model dimension of 384 and 3072-D feedforward layers, 4-headed self attention, and a left context of 31 tokens. Overall, the LM rescorer has 71M parameters. The LM is trained using the same text-only data used to train the BERT text encoder. During inference, the conformer LM is used to rescore the lattice after the non-causal cascaded encoders. Compared to LM rescoring in Table 5, deliberation with JATD (`E1`) performs similarly to LM rescoring (`B2`) in long-tail words, and 6% and 12% relatively better for VS and SxS test sets, respectively. Note that without BERT encoder the deliberation rescorer size of `E1` is 57M, which is 20% smaller than LM (71M). When incorporating BERT and semi-supervised training, we achieve more significant and uniform improvements: VS (8.9%), long-tail (1.8%), and SxS test set (15.6%), all in relative WER reductions.

Table 5: *WER (%) improvements by deliberation rescoring using text-only and semi-supervised training.*

Model		# Text Enc. layer	WER (%)		
			VS	SxS	LT
B0	Cas. Enc. [2]	-	5.4	30.2	31.6
B1	Deliberation	2L	4.9	27.6	30.7
E1	Delib. JATD-All		5.0	26.9	27.8
E2	+ 12L-BERT	12L	4.7	25.9	27.9
E3	+ Semi-sup		**4.7**	**25.8**	**27.1**
B2	LM Rescoring	12L	5.3	30.6	27.6

4.2. Side-by-Side Comparison with LM Rescoring

We further compare the proposed deliberation model to a state-of-the-art LM rescorer [42] in a decoding setup with endpointing. The baseline cascaded encoders in [42] consist of a small causal encoder and large non-causal encoder. The causal encoder consists of a 7-layer conformer and the non-causal encoder has a 10-layer right-context conformer with an overall right-context of 0.9s. The encoder output dimension is projected to 384 to reduce model size. Following [1], we use the hybrid autoregressive transducer (HAT) version of the LM rescoring model (the non-HAT version performs worse). For deliberation, we take the best-performing rescorer (`E3` in Table 5) and reduce the decoder dimension to 384 to match the cascaded encoder. The deliberation rescorer has a total size of 106M. We found that a non-HAT decoder works better for deliberation than the HAT version.

We compare deliberation and LM rescoring in a human side-by-side evaluation. A total of 705 utterances are transcribed by both models, and are sent to two human transcribers to rate. Each transcript is rated as either a win for deliberation over LM rescoring (only deliberation is correct), or a loss (only LM rescoring is correct), or neutral (both models are correct or incorrect). Table 6 shows the deliberation rescorer changes 9% of traffic, and has significantly more wins (114) than losses (50) compared to LM rescoring. Overall, the p-Value of $< 0.1\%$ shows the difference is statistically significant.

Table 6: *Side-by-side eval: LM vs. Delib. Rescoring*

Changed (%)	Win	Loss	Neutral	p-Value
9.0	114	50	541	<0.1%

In terms of VS WERs in this comparison, the deliberation model achieves a WER of 5.4%. This is 10% relative better than cascaded encoders (6.0%), and 8% better than HAT LM rescoring (5.9%). We have also tried increasing the LM size to around 100M or using a BERT LM but did not see any improvement. The deliberation model achieves an EP50 (median latency) of 380 ms and EP90 (90th latency) of 720 ms, similar to LM rescoring.

5. Conclusion

We researched text-only and semi-supervised training for LAS-based deliberation. By incorporating pretrained BERT text encoder, large-scale JATD and semi-supervised training, we have improved the deliberation performance by 4% for VS, and 12% relative for long-tail in terms of WERs. In the latest cascaded encoder setup with endponting, we show the proposed deliberation rescorer outperforms a state-of-the-art LM rescoring method by 8% relative in terms of VS WER, and wins in a human side-by-side evaluation.

6. References

[1] T. N. Sainath, Y. He, A. Narayanan, R. Botros, R. Pang, D. Rybach, C. Allauzen, E. Variani, J. Qin, Q.-N. Le-The, S.-Y. Chang, B. Li, A. Gulati, C.-C. Yu, Jiahui Chiu, D. Caseiro, W. Li, Q. Liang, P. Rondo *et al.*, "An efficient streaming non-recurrent on-device end-to-end model with improvements to rare-word modeling," *Interspeech*, 2021.

[2] A. Narayanan, T. N. Sainath, R. Pang, J. Yu, C.-C. Chiu, R. Prabhavalkar, E. Variani, and T. Strohman, "Cascaded encoders for unifying streaming and non-streaming asr," in *IEEE ICASSP*, 2021, pp. 5629–5633.

[3] 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 *IEEE ICASSP*, 2021, pp. 5904–5908.

[4] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao *et al.*, "Developing RNN-T models surpassing high-performance hybrid models with customization capability," *arXiv preprint arXiv:2007.15188*, 2020.

[5] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[6] X. Wang, Z. Yao, X. Shi, and L. Xie, "Cascade RNN-transducer: Syllable based streaming on-device Mandarin speech recognition with a syllable-to-character converter," in *SLT*, 2021, pp. 15–21.

[7] G. Saon, Z. Tüske, D. Bolanos, and B. Kingsbury, "Advancing RNN transducer technology for speech recognition," in *IEEE ICASSP*, 2021, pp. 5654–5658.

[8] J. Li, "Recent advances in end-to-end automatic speech recognition," *arXiv preprint arXiv:2111.01690*, 2021.

[9] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *IEEE ICASSP*, 2020, pp. 7799–7803.

[10] K. Hu, R. Pang, T. N. Sainath, and T. Strohman, "Transformer based deliberation for two-pass speech recognition," in *SLT*, 2021, pp. 68–74.

[11] S. Mavandadi, T. N. Sainath, K. Hu, and Z. Wu, "A deliberation-based joint acoustic and text decoder," in *Interspeech*, 2021.

[12] J. Shin, Y. Lee, and K. Jung, "Effective sentence scoring method using BERT for speech recognition," in *ACML*. PMLR, 2019, pp. 1081–1093.

[13] D. Fohr and I. Illina, "BERT-based semantic model for rescoring n-best speech recognition list," in *INTERSPEECH*, 2021.

[14] O. Hrinchuk, M. Popova, and B. Ginsburg, "Correction of automatic speech recognition with transformer sequence-to-sequence model," in *IEEE ICASSP*, 2020, pp. 7074–7078.

[15] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," in *Proc. IEEE ICASSP*, 2019, pp. 5651–5655.

[16] Y. Tang, J. Pino, C. Wang, X. Ma, and D. Genzel, "A general multi-task learning framework to leverage text data for speech to text tasks," in *IEEE ICASSP*, 2021, pp. 6209–6213.

[17] K. Deng, S. Cao, Y. Zhang, and L. Ma, "Improving hybrid CTC/attention end-to-end speech recognition with pretrained acoustic and language model," *arXiv preprint arXiv:2112.07254*, 2021.

[18] A. Bapna, Y.-a. Chung, N. Wu, A. Gulati, Y. Jia, J. H. Clark, M. Johnson, J. Riesa, A. Conneau, and Y. Zhang, "SLAM: A unified encoder for speech and language modeling via speech-text joint pre-training," *arXiv preprint arXiv:2110.10329*, 2021.

[19] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end ASR," *arXiv preprint arXiv:1803.10299*, 2018.

[20] P. Bahar, T. Bieschke, and H. Ney, "A comparative study on end-to-end speech to text translation," in *ASRU*. IEEE, 2019, pp. 792–799.

[21] T. N. Sainath, R. Pang, R. J. Weiss, Y. He, C.-c. Chiu, and T. Strohman, "An attention-based joint acoustic and text on-device end-to-end model," in *IEEE ICASSP*, 2020, pp. 7039–7043.

[22] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[23] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[24] Y. Zhang, D. S. Park, W. Han, J. Qin, A. Gulati, J. Shor, A. Jansen, Y. Xu, Y. Huang, S. Wang *et al.*, "BigSSL: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2109.13226*, 2021.

[25] T. Doutre, W. Han, M. Ma, Z. Lu, C.-C. Chiu, R. Pang, A. Narayanan, A. Misra, Y. Zhang, and L. Cao, "Improving streaming automatic speech recognition with non-streaming model distillation on unsupervised data," in *IEEE ICASSP 2021*, 2021, pp. 6558–6562.

[26] D. Hwang, A. Misra, Z. Huo, N. Siddhartha, S. Garg, D. Qiu, K. C. Sim, T. Strohman, F. Beaufays, and Y. He, "Large-scale asr domain adaptation using self-and semi-supervised learning," *arXiv preprint arXiv:2110.00165*, 2021.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[28] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[29] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, "Transducer-based streaming deliberation for cascaded encoders," in *IEEE ICASSP 2022 (to appear)*.

[30] X. Gonzalvo, S. Tazari, C.-a. Chan, M. Becker, A. Gutkin, and H. Silen, "Recent advances in google real-time HMM-driven unit selection synthesizer," *Interspeech*, 2016.

[31] G. Pundak and T. Sainath, "Lower frame rate neural network acoustic models," in *Proc. Interspeech 2016*, 2016, pp. 22–26.

[32] 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, pp. 920–927.

[33] B. Li, A. Gulati, J. Yu, T. N. Sainath, C.-C. Chiu, A. Narayanan, S.-Y. Chang, R. Pang, Y. He, J. Qin *et al.*, "A better and faster end-to-end model for streaming ASR," in *IEEE ICASSP*, 2021, pp. 5634–5638.

[34] R. Botros, T. N. Sainath, R. David, E. Guzman, W. Li, and Y. He, "Tied & reduced RNN-T decoder," *arXiv preprint arXiv:2109.07513*, 2021.

[35] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in *Proc. IEEE ICASSP*, 2012, pp. 5149–5152.

[36] H. Liao, E. McDermott, and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription," in *ASRU*, 2013, pp. 368–373.

[37] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. N. Sainath, and M. Bacchiani, "Generation 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, pp. 379–383.

[38] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[39] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[40] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu, I. McGraw, and C.-C. Chiu, "Two-pass end-to-end speech recognition," in *Proc. Interspeech*, 2019, pp. 2773–2777.

[41] 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," *arXiv preprint arXiv:2008.10491*, 2020.

[42] T. N. Sainath, Y. R. He, A. Narayanan, R. Botros, W. Wang, D. Qiu, C.-C. Chiu, R. Prabhavalkar, A. Gruenstein, A. Gulati, B. Li, D. Rybach, E. Guzman, I. McGraw, J. Qin, K. Choromanski, Q. Liang, R. David, R. Pang, S.-y. Chang, T. Strohman, W. R. Huang, W. Han, Y. Wu, and Y. Zhang, "Improving the latency and quality of cascaded encoders," *IEEE ICASSP*, 2022 (to appear).