INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

A Deliberation-based Joint Acoustic and Text Decoder

Sepand Mavandadi, Tara N. Sainath, Ke Hu, Zelin Wu

Google Inc., U.S.A

{sepand, tsainath, huk, zelinwu}@google.com

Abstract

We propose a new two-pass E2E speech recognition model that improves ASR performance by training on a combination of paired data and unpaired text data. Previously, the joint acoustic and text decoder (JATD) has shown promising results through the use of text data during model training and the recently introduced deliberation architecture has reduced recognition errors by leveraging first-pass decoding results. Our method, dubbed Deliberation-JATD, combines the spelling correcting abilities of deliberation with JATD's use of unpaired text data to further improve performance. The proposed model produces substantial gains across multiple test sets, especially those focused on rare words, where it reduces word error rate (WER) by between 12% and 22.5% relative. This is done without increasing model size or requiring multi-stage training, making Deliberation-JATD an efficient candidate for on-device applications.

1. Introduction

E2E models [1, 2, 3, 4, 5, 6, 7, 8] combine the acoustic (AM), pronunciation (PM) and language models (LM) of a conventional ASR system into a single neural network. This structure makes them significantly smaller than conventional models [2, 9] and ideal for on-device ASR [1]. However, the performance of E2E models on rare word recognition still lags behind conventional models. The performance gap is partially because they lack the ability to train using text-only data, which is abundant and often utilized by conventional LMs.

There have been multiple approaches for augmenting E2E models and training procedures to incorporate unpaired text data. Broadly speaking, these approaches use some combination of an LM trained on text data (shallow, cold, deep fusion [10, 11, 12, 13]) and a multi-stage training procedure that incorporates unpaired data ("weak distillation" [14], "back-translation" [15], "cycle-consistency" [16, 17, 18]). Each approach produces improvements in performance, but also increases some combination of model size, training and inference complexity, making it less desirable for on-device applications.

A recent approach, the joint acoustic and text decoder (JATD) [19, 20] side-steps these issues. It utilizes unpaired data during E2E model training either directly or by generating the missing half of the data; using TTS to generate audio from text and ASR to generate text from audio. Previous methods using unpaired data in training have shown limited success in improving performance on real audio test sets [14, 21]. JATD produces stronger results by using a fixed context vector as a "domain ID" to distinguish between paired and unpaired data during training. Paired data is processed as normal with the encoder computing an acoustic context vector that is fed to the decoder. For unpaired data (text-only or with synthesized audio), JATD bypasses the encoder network by using a fixed but learnable context vector in place of the encoder output, allowing the model to train on text-only data and avoiding the encoder training on synthesized audio.

The JATD architecture results in only a trivial increase in model size. It has been implemented within a LAS two-pass decoding framework and trained on both audio-text pair data and unpaired text-only data (by synthesizing into TTS utterances), showing substantial improvements in WER, especially on rare words [19].

Beside data augmentation approaches, novel model architectures also show improvements in recognizing rare words. Recently, deliberation models [22, 23, 24] have used an attention-based two-pass design to achieve state-of-the-art performance on Google VoiceSearch test sets. Similar to other two-pass models, deliberation uses its first-pass decoder to produce streaming hypotheses and its second-pass decoder to attend to the first-pass hypotheses alongside the encoder outputs for re-decoding or rescoring. The hypothesis attention allows deliberation to act like a spelling corrector on full-context first-pass hypotheses. This results in substantial performance gains, especially on rare words [23, 24].

Our novel contribution is to combine the text-only training capabilities of JATD with the spelling-correction benefits of deliberation. Our approach, dubbed deliberation-JATD, augments deliberation's attention contexts to use JATD's fixed context vectors, enabling the architecture to train on text-only data. Experiments show deliberation-JATD improving rare word performance by at least 12% relative to both LAS-JATD and deliberation without any degradation on VoiceSearch tasks.

2. Methods

In this section, we describe our baseline approaches, deliberation and LAS-JATD, as well as the proposed Deliberation-JATD method.

2.1. Baseline Approaches

2.1.1. Deliberation

Deliberation networks (fig. 1) consist of a shared encoder, a first-pass RNN-T decoder, and a second-pass deliberation decoder. The shared encoder takes log-mel filterbank energies, $\mathbf{x} = (\mathbf{x}_1 \dots \mathbf{x}_T)$, where T denotes the number of frames, and generates an encoding, $\mathbf{e}$. This encoder output, $\mathbf{e}$, is then fed to an RNN-T decoder to produce first-pass decoding results, $\mathbf{y}^r$, in a streaming fashion. The deliberation decoder attends to both $\mathbf{e}$ and $\mathbf{y}^r$, producing two context vectors, $\mathbf{c}_e^a$ and $\mathbf{c}_b^a$, that are concatenated and passed as inputs to an attention-based LSTM decoder. This decoder produces the final probabilities, $\mathbf{y}^d$, which can be written as $p(\mathbf{y}^d|\mathbf{x}, \mathbf{c}_e^a, \mathbf{c}_b^a, \mathbf{y}_{u-1:1}^d)$, where $\mathbf{y}_{u-1:1}^d = \mathbf{y}_{u-1}^d, \dots, \mathbf{y}_1^d$ indicates all previous decoded labels of a single hypothesis during inference.

Inference for deliberation models is done in two passes. First, the RNN-T decoder processes encoder outputs, $\mathbf{e}$, to produce the first-pass sequence, $\mathbf{y}^r$. Then, the deliberation decoder attends to $\mathbf{e}$ and the complete first pass hypotheses, $\mathbf{y}^r$, and per-

http://dx.doi.org/10.21437/Interspeech.2021-165

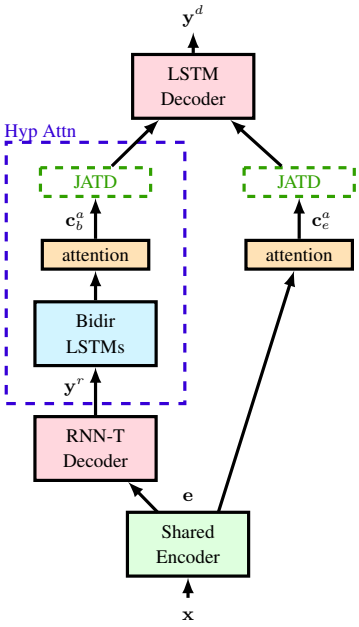

Figure 1: *2-pass model architecture. With "Hyp Attn" block: a deliberation model. Without "Hyp Attn" block: a 2-pass LAS model. Including the JATD block only on the right branch results in the "partial" variant of Deliberation-JATD. Including both JATD blocks results in the "full" variant.*

forms a second beam search to generate $\mathbf{y}^d$. This second pass acts as a spell-corrector, using the full context of the first pass hypothesis to substantially improve performance [23].

Deliberation training requires audio-text pairs and does not offer a natural way to incorporate unpaired text data. In the following section, we describe JATD, which addresses this shortfall.

2.1.2. LAS-JATD

JATD was implemented [19] on a two-pass LAS model running beam search in the second-pass decoder. This two-pass LAS model can be succinctly described as a deliberation model where the second-pass LSTM decoder does not use the the first-pass RNN-T decoder outputs, $\mathbf{y}^r$, and only attends to the shared encoder outputs, $\mathbf{e}$. This is equivalent to fig. 1 with the "Hyp Attn" block removed. The final log probabilities output by LAS can be written as $\log p(\mathbf{y}_u^d|\mathbf{x}^a, \mathbf{c}_e^a, \mathbf{y}_{u-1:1}^d)$.

LAS-JATD [19] augments LAS to enable training on paired audio-text data, as well as unpaired (i.e. text-only) data. This is done through the introduction of a new learnable fixed context vector, $\mathbf{c}_e^l$, which is used as an alternative to the acoustic context vector $\mathbf{c}_e^a$ (fig. 2). During inference, two log probabilities are produced, one based on $\mathbf{c}_e^a$ and the other based on $\mathbf{c}_e^l$. These are interpolated using weight λ to produce the final output log probabilities:

$$\lambda \log p(\mathbf{y}_u^d|\mathbf{x}, \mathbf{c}_e^a, \mathbf{y}_{u-1:1}^d) + (1-\lambda) \log p(\mathbf{y}_u^d|\mathbf{c}_e^l, \mathbf{y}_{u-1:1}^d) \quad (1)$$

The first term in this equation represents the familiar acoustic model (i.e. a regular two-pass LAS). The second term, $\log p(\mathbf{y}_u^d|\mathbf{c}_e^l, \mathbf{y}_{u-1:1}^d)$, can be thought of as a language model since it does not depend on acoustic features, $\mathbf{x}$.

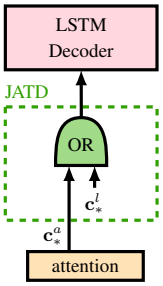

Figure 2: *Details of JATD implementation. The "JATD" block can be added to the outputs of either attention block in fig. 1.*

LAS-JATD also provides a framework for incorporating unpaired data into training (eq. 2). It uses both acoustic and learnable context vectors when training on both paired and unpaired data. Acoustic context vectors are generated based on real audio, $\mathbf{x}^a \in \mathbf{x}$, for paired examples and "created" audio, $\mathbf{x}^l \in \mathbf{x}$, for unpaired examples. Importantly, it restricts training so only paired examples update the encoder attention parameters and only unpaired examples update the fixed context vector. This avoids biasing acoustic attention parameters towards unpaired data, and was found to be effective in [19]. In this work, we explore synthesizing audio based on text data to create $\mathbf{x}^l$.

$$\mathcal{L} = \begin{cases} \lambda \log p(\mathbf{y}_u^d|\mathbf{x}^a, \mathbf{c}_e^a, \mathbf{y}_{u-1:1}^d) + (1-\lambda) \log p(\mathbf{y}_u^d|\mathbf{c}_e^l, \mathbf{y}_{u-1:1}^d) \\ \text{if paired example} \\ \lambda \log p(\mathbf{y}_u^d|\mathbf{x}^l, \mathbf{c}_e^a, \mathbf{y}_{u-1:1}^d) + (1-\lambda) \log p(\mathbf{y}_u^d|\mathbf{c}_e^l, \mathbf{y}_{u-1:1}^d) \\ \text{if unpaired example} \end{cases} \quad (2)$$

LAS-JATD improves performance through the addition of unpaired data to its training set, but misses out on gains from the spell-correcting capabilities of deliberation models.

2.2. Proposed Method: Deliberation-JATD

We propose the Deliberation-JATD model, combining deliberation's spell-correcting benefits with JATD's ability to train on unpaired data. Similar to LAS-JATD, this model uses fixed context vectors as an alternative to attention context vectors, c_e^a and c_b^a. This results in a new set of output log probabilities that act as a language model.

Given that deliberation models contain two attention contexts, we examine two Deliberation-JATD variations. The first, dubbed "partial JATD", uses the fixed context vector $\mathbf{c}_e^l$ as an alternative to the encoder attention context, $\mathbf{c}_e^a$, while continuing to use first-pass decoder context, $\mathbf{c}_b^a$. This results in the LM log probabilities, $\log p(\mathbf{y}_u^d|\mathbf{x}, \mathbf{c}_e^l, \mathbf{c}_b^a, \mathbf{y}_{u-1:1}^d)$, and the following final model outputs used during inference:

$$\lambda \log p(\mathbf{y}_u^d|\mathbf{x}, \mathbf{c}_e^a, \mathbf{c}_b^a, \mathbf{y}_{u-1:1}^d) + (1-\lambda) \log p(\mathbf{y}_u^d|\mathbf{x}, \mathbf{c}_e^l, \mathbf{c}_b^a, \mathbf{y}_{u-1:1}^d) \quad (3)$$

The second variant, named "full JATD", goes one step further and adds a second fixed context vector, $\mathbf{c}_b^l$, as an alternative to its first pass decoder attention context, $\mathbf{c}_b^a$ (eq. 4). This means that both attention contexts are replaced and the LM log probabilities having no dependence on the acoustic inputs, $\mathbf{x}$.

$$\lambda \log p(\mathbf{y}_u^d|\mathbf{x}, \mathbf{c}_e^a, \mathbf{c}_b^a, \mathbf{y}_{u-1:1}^d) + (1-\lambda) \log p(\mathbf{y}_u^d|\mathbf{c}_e^l, \mathbf{c}_b^l, \mathbf{y}_{u-1:1}^d) \quad (4)$$

We use the "joint" training strategy [19] to train both variants on a mix of supervised audio and text-only data paired with TTS audio. The resulting training loss is similar to eq. 2, but incorporates eq. 3 or eq. 4 for the unpaired text term, depending on the model variation used. Similar to LAS-JATD, we distinguish between real audio from paired data, $\mathbf{x}^a$, and synthetic audio from text data, $\mathbf{x}^l$, and only backpropagate some parameters for each type of data. Specifically, encoder attention parameters are held constant for synthetic audio and the fixed attention context vectors are held constant for real audio. The second-pass decoder parameters are updated for both types of data.

Full JATD's LM is similar to LAS-JATD's LM in that it replaces all attention contexts (from the encoder and first-pass decoder) with fixed vectors. This means that the LM component's second-pass decoder completely ignores the first-pass decoder and all acoustic information. As a result, the LM learns to spell-correct based purely on previous model outputs, $\mathbf{y}_{u-1:1}^d$. In contrast, partial JATD was designed to more frequently expose the second-pass decoder to the first-pass decoder during training. Its LM keeps the first-pass decoder attention context, allowing the second-pass decoder to train with it and the LM to make use of some indirect acoustic information.

2.3. Training

All models were initialized from an RNN-T trained with supervised audio-text paired data. The same model was also used as the baseline RNN-T model when analyzing performance. All training sets that included TTS used a mix of 90% audio-text pairs and 10% pure text data with corresponding TTS. We will refer to this training set as the "mixed audio" training set. For all JATD models, an interpolation weight of $\lambda = 0.1$ was found to work well in training.

3. Experiment Details

3.1. Training Sets

We use the same paired audio-text training set as [25]. This dataset consists of approximately 180M multi-domain utterances spanning domains of search, farfield, telephony and YouTube English utterances. The search and farfield utterances are anonymized and hand-transcribed and are representative of Google's voice search traffic.

Our unpaired data consisted of 4.6M samples of query text from anonymized Google Maps traffic. We paired this text with synthesized audio generated by a multi-speaker TTS system based on the architecture described in [26]. Our synthesized audio is generated in the voice of 98 English speakers covering American, Australian, British and Singaporean accents with a Google Assistant clean speaking style. Each utterance's audio was synthesized using a single voice assigned randomly during training set generation.

Both real and synthesized audio training data were artificially corrupted using a room simulator. Various degrees of noise and reverberation were added such that the overall SNR is between 0dB and 30dB, with an average SNR of 12dB [27]. The noise sources were from YouTube and noisy environmental recordings.

3.2. Test Sets

We use two primary evaluation sets: "VS" and "SXS". VS consists of 14k anonymized hand-transcribed utterances from Google traffic. SXS is a sample of 1,200 real-audio utterances where the conventional model [28] outperformed the E2E LAS rescoring model [9]. This dataset consists largely of rare words and is useful for measuring how including text-only training data can improve some of these errors.

We used a corpus composed purely of rare word utterances [29] to specifically evaluate model performance on rare nouns. This corpus is a non-overlapping random subset of the same text data that was sampled to create the unpaired training data. It consists of utterances containing words that (1) occurred either once or not at all in the paired training set and (2) accounted for less than 1 in every million words in the text data. For privacy reasons, we only included words that occurred more than 1000 times in the text data.

We created two evaluation sets derived from this rare word corpus. The "TTS" set is a subset of 10,000 of these utterances combined with synthetic audio generated by the same system that synthesized audio for the training set in [26]. For this test set, the audio synthesizing system was configured to use a voice profile distinct from all those used in the training set. The "Spoken Text" evaluation set is a random sample of 200 utterances from the TTS set, but manually spoken and recorded by one of our team members. This set was used to verify that improvements in model performance on the TTS evaluation sets were matched on real audio.

3.3. Architecture Details

We used the same deliberation model configuration as [23]. All experiments used 128-dimensional log-Mel features, computed with a 32ms window and shifted every 10ms. Similar to [25], features for each frame are stacked with 3 frames to the left and then downsampled by 3 to a 30ms frame rate.

The first-pass RNN-T network is similar to [9], consisting of an 8 LSTM layer encoder and 2 LSTM layer prediction network. Each LSTM layer has 2,048 hidden units followed by a 640-dimensional projection layer. There is a factor of 2 time-reduction layer after the second encoder LSTM layer. The outputs of encoder and prediction network are fed to a 640 hidden unit joint network followed by a softmax layer predicting 4,096 lowercase wordpieces.

The first-pass RNN-T hypotheses are padded with end-of-sentence label $\langle \backslash s \rangle$ to a length of 120. Each subword in a hypothesis is mapped to a vector by a 96-dimensional embedding layer and encoded by a 2-layer bidirectional LSTM encoder, where each layer has 2,048 hidden units followed by a 320-dimensional projection. Both attention models use multi-headed attention [30] with four attention heads. The two output context vectors are concatenated and fed to a 2-layer LSTM decoder (2,048 hidden units followed by a 640-dimensional projection per layer). The second-pass attention decoder has a 4,096-dimensional softmax layer to predict the same mixed-case wordpieces [31] as the RNN-T. We use a similar architecture for our LAS models. The second-pass decoder in these models is the same as before (2 LSTM layers, each with 2,048 hidden units followed by a 640-dimensional projection).

The total size of the RNN-T model is 114M parameters, and the second-pass decoder has 33M parameters. All models are trained in Tensorflow [32] using the Lingvo [33] toolkit on a v2-128 Cloud TPU slice with a global batch size of 4,096.

For JATD models, the interpolation weight, λ, for inference

Table 1: *Model performance comparison. Lowest baseline WER values are underlined, lowest overall WER values are* ***bolded****.*

ID	Model	Training Data	WER (%)			
			VS	SXS	TTS	Spoken Text
B0	RNN-T	Paired	6.6	28.3	41.9	26.9
B1	RNN-T	Mixed	6.2	30.4	39.6	22.5
B2	LAS	Paired	6.7	25.6	41.4	27.4
B3	LAS	Mixed	6.7	24.6	38.4	24.1
B4	LAS-JATD	Mixed	7.0	26.1	33.1	24.0
B5	Deliberation	Paired	**5.7**	22.0	39.5	22.2
B6	Deliberation	Mixed	5.8	21.9	34.8	20.5
E0	Deliberation-JATD (Partial)	Mixed	5.8	21.9	30.9	20.7
E1	Deliberation-JATD (Full)	Mixed	**5.7**	**21.7**	**30.6**	**18.7**

is chosen (between 0.01, 0.025, and 0.05) to optimize WER for the VS test set. Results shown in table 1 use $\lambda = 0.025$ for Deliberation-JATD and $\lambda = 0.01$ for LAS-JATD.

4. Results

We now analyze the performance of our Deliberation-JATD model. Table 1 compares the performance of our two model variants (E0, E1) with a set of baseline models (B0 to B6). All models are trained on either paired data or the mixed audio training set (described in section 2.3).

B0 is an RNN-T model and serves as a baseline trained on paired data. It is also the model we used to initialize all LAS and deliberation variants. B1 has the same architecture as B0, but trained from scratch on the mixed audio training set. The addition of TTS data to training improves RNN-T performance on the TTS, VS, and Spoken Text test sets while degrading on the SXS set.

B2 is an LAS model trained on paired data and B3 is the same model trained on the mixed audio training set. B4 is our LAS-JATD implementation. Training on mixed audio (B3) provides some modest performance improvements compared to B2. The LAS-JATD model shows the lowest WER among baseline models on the TTS set. It also improves on the Spoken Text relative to regular LAS, but degrades VS and SXS performance.

B5 and B6 are implementations of the two-pass deliberation model trained on the paired and mixed audio training sets, respectively. They show the strongest metrics on Spoken Text as well as the VS and SXS sets. They also have the lowest TTS WER among non-JATD models. Training deliberation on the mixed audio (B6), as opposed to only the paired data (B5), results in significant improvements on all but the VS test set.

We compare the aforementioned baselines against our Deliberation-JATD models: the full variant (E0) and the partial variant (E1), both trained on the mixed audio training set. Both variants show significant gains on all sets sets aside from VS, which is roughly unchanged. The full variant (E1) produces the lowest WER of all models on the SXS, TTS, and Spoken Text test sets and matches the lowest WER obtained on VS by B5. On the TTS test set, it shows a 22.5% improvement relative to deliberation trained on paired data (B5), and a 12% improvement relative to deliberation trained on mixed data (B6). Similar gains are seen on the Spoken Text set.

Comparing the Deliberation-JATD models, we notice that the full variant outperforms the partial variant despite the fact that the full JATD LM term (described in section 2.2) ignores the first-pass decoder outputs, while the partial JATD LM term uses them. We speculate that partial JATD would benefit from a real/TTS bit passed to the bidirectional LSTMs that encode the first-pass decoder output. This would allow its LM component to distinguish between paired and TTS audio. We leave this as future work.

Table 2: *Sample wins and losses comparing full deliberation-JATD (E1) and deliberation (B5) on the Spoken Text test set. Correct and incorrect portions highlighted in green and red, respectively.*

Deliberation (B5)	Full Deliberation-JATD (E1)
tough trees leasing office	**toftrees** leasing office
chow mein jackson mississippi	**cal-maine** jackson mississippi
distance from **wanderleo** to punta cana	distance from **juan dolio** to punta cana
nellis ford realty	**nellysford** realty
southline	**south lyon**
the mansions of **shadowbriar** houston texas	the mansions of **shadow briar** houston texas
delias near me	**delia's** near me

Finally, Table 2 shows a sample of wins and losses when comparing deliberation (B5) to the the full variant of deliberation-JATD (E2). The deliberation-JATD wins mostly by correcting transcription errors for proper nouns such as "toftrees" and "nellysford". The losses are sometimes also related to proper nouns (e.g. "southline" to "south lyon"), but mostly due to spelling errors, e.g. "delia's" in place of "delias".

5. Conclusions

We presented a new Deliberation-JATD model, which incorporates unpaired text data in a deliberation model training jointly with acoustic (i.e. paired) data. The proposed method significantly outperforms both Deliberation and LAS-JATD models, reducing WER by up to 22.5% relative to a regular deliberation model [23] on a rare word test set. Although the regular deliberation is improved by training from scratch using both paired and unpaired data, it still lags behind the Deliberation-JATD model by 12% in terms of WER. The superior performance of Deliberation-JATD is achieved without additional inference complexity, mult-stage training, or performance degradation on Google Voice Search tasks.

6. Acknowledgements

Thank you to Ruoming Pang and Cal Peyser with their help in producing training and test sets used for this work.

7. References

[1] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming End-to-end Speech Recognition For Mobile Devices," in *Proc. ICASSP*, 2019.

[2] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, N. Jaitly, B. Li, and J. Chorowski, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[3] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.

[4] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep neural networks," in *Proc. ICASSP*, 2012.

[5] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *Proc. ASRU*, 2017, pp. 193–199.

[6] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[7] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[8] C.-C. Chiu and C. Raffel, "Monotonic chunkwise alignments," in *Proc. ICLR*, 2017.

[9] 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.

[10] J. K. Chorowski and N. Jaitly, "Towards Better Decoding and Language Model Integration in Sequence to Sequence Models," in *Proc. Interspeech*, 2017.

[11] A. Sriram, H. Jun, S. Sateesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," *CoRR*, vol. abs/1708.06426, 2017.

[12] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *Proc. ICASSP*, 2018.

[13] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of language-independent end-to-end asr with language model fusion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6096–6100.

[14] B. Li, T. N. Sainath, R. Pang, and Z. Wu, "Semi-supervised Training for End-to-End Models Via Weak Distillation," in *Proc. ICASSP*, 2019.

[15] R. Sennrich, B. Haddow, and A. Birch, "Improving Neural Machine Translation Models with Monolingual Data," in *ACL*, 2016.

[16] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. Le Roux, "Cycle-Consistency Training for End-to-End Speech Recognition," in *Proc. ICASSP*, May 2019, pp. 6271–6275.

[17] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Almost unsupervised text to speech and automatic speech recognition," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019, pp. 5410–5419.

[18] Y. Bai, J. Yi, J. Tao, Z. Tian, and Z. Wen, "Learn Spelling from Teachers: Transferring Knowledge from Language Models to Sequence-to-Sequence Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 3795–3799.

[19] T. N. Sainath, R. Pang, R. J. Weiss, Y. He, C. Chiu, and T. Strohman, "An attention-based joint acoustic and text on-device end-to-end model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7039–7043.

[20] P. Wang, T. N. Sainath, and R. J. Weiss, "Multitask training with text data for end-to-end speech recognition," *arXiv preprint arXiv:2010.14318*, 2020.

[21] D. Zhao, T. N. Sainath, D. Rybach, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-End Contextual Biasing," in *submitted to Proc. Interspeech*, 2019.

[22] Y. Xia, F. Tian, L. Wu, J. Lin, T. Qin, N. Yu, and T. Liu, "Deliberation networks: Sequence generation beyond one-pass decoding," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1782–1792.

[23] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7799–7803.

[24] K. Hu, R. Pang, T. N. Sainath, and T. Strohman, "Transformer based deliberation for two-pass speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, to appear.

[25] 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 *to appear in Proc. ASRU*, 2019.

[26] C. Peyser, H. Zhang, T. N. Sainath, and Z. Wu, "Improving performance of end-to-end asr on numeric sequences," *Proc. Interspeech 2019*, pp. 2185–2189, 2019.

[27] 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.

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

[29] 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," .

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, 2017.

[31] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5149–5152.

[32] M. Abadi et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," Available online: http://download.tensorflow.org/paper/whitepaper2015.pdf, 2015.

[33] J. Shen, P. Nguyen, Y. Wu, Z. Chen, et al., "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," 2019.