SpeechStew: Simply Mix All Available Speech Recognition Data to Train One Large Neural Network

William Chan, Daniel S. Park, Chris A. Lee, Yu Zhang, Quoc V. Le, Mohammad Norouzi

Google Research, Brain Team

{williamchan,danielspark,chrisalee,ngyuzh,qvl,mnorouzi}@google.com

Abstract

We present SpeechStew, a speech recognition model that is trained on a combination of various publicly available speech recognition datasets: AMI, Broadcast News, Common Voice, LibriSpeech, Switchboard/Fisher, Tedlium, and Wall Street Journal. SpeechStew simply mixes all of these datasets together, without any special re-weighting or re-balancing of the datasets. SpeechStew achieves SoTA or near SoTA results across a variety of tasks, without the use of an external language model. Our results include 9.0% WER on AMI-IHM, 4.7% WER on Switchboard, 8.3% WER on CallHome, and 1.3% on WSJ, which significantly outperforms prior work with strong external language models. We also demonstrate that SpeechStew learns powerful transfer learning representations. We fine-tune SpeechStew on a noisy low resource speech dataset, CHiME-6. We achieve 38.9% WER without a language model, which compares to 38.6% WER to a strong HMM baseline with a language model.

Index Terms: end-to-end speech recognition, multi-domain speech recognition

1. Introduction

End-to-end speech recognition models [1, 2, 3] have seen remarkable success in recent years [4]. For instance, end-to-end speech recognition models have achieved state-of-the-art (SoTA) results on LibriSpeech [5] and large proprietary datasets [6]. The success of these methods have often been attributed to the abundance of training data [6] and the use of large deep models [5]. However, on noisy, low resource speech recognition datasets, such as CHiME-6 [7], where overfitting is a significant problem, end-to-end methods tend to struggle relative to HMM-based baselines [8]. For example, the best previously published end-to-end model achieved 49.0% WER on the CHiME-6 dev set [8], while the best HMM model achieves 36.9% WER [9].

Multi-lingual training [10, 11], multi-domain training [12, 13], unsupervised pre-training [14, 15], semi-supervised learning [16, 17] and transfer learning [18, 19] are some techniques proposed in the literature to enhance generalization. These methods optimize speech recognition models on data from related tasks (typically of high resource), to help the specific task of interest (typically of low resource). For example, in multi-lingual training, the knowledge from a high resource language may transfer to a low resource language [20]. In multi-domain training, combining different domain datasets of the same language, could facilitate the cross-sharing of knowledge across the domains [13]. In unsupervised pre-training, the knowledge from the pre-training task may transfer to the supervised task [21]. In transfer learning, a general model is trained with a large amount of data. Subsequently, its knowledge is transferred via fine-tuning on training data from a downstream task that is typically of low resource [22].

Neural networks have benefited from simply scaling up to larger and deeper models across multiple application domains like computer vision [23, 24] and language modelling [25, 26]. It has been observed that these large models tend to enhance generalization [27]. End-to-end speech recognition models have also been benefited from the increases in model capacity [28, 29, 5].

This paper presents SpeechStew. SpeechStew is a simple approach to end-to-end speech recognition, which leverages both multi-domain training and transfer learning. SpeechStew follows the following simple recipe:

1. Combine all available speech recognition data without any domain-dependent re-balancing or re-weighting.
2. Train a single large neural network (a 100M or 1B parameter model) on the combined data.

Our method does not utilize any domain labels, or introduce any additional hyperparameters for combining the data. We do not incorporate an external language model during inference, yet our result compares favourably to prior work that utilize strong language models, achieving SoTA or near SoTA results across various tasks (AMI, Common Voice, LibriSpeech, Switchboard, Tedlium, and WSJ).

We also demonstrate that SpeechStew has strong transfer learning capabilities. When presented with a new unseen low resource dataset (CHiME-6 in our setup), we merely:

3. Fine-tune SpeechStew on the new labelled dataset.

We find that this straightforward pre-training and fine-tuning procedure yields near-SoTA results on CHiME-6. This is encouraging since CHiME-6 is a particularly challenging task [7] for end-to-end speech recognition models, which suffer from over-fitting issues [8]. We also demonstrate that our method is complementary to other pre-training methods, in particular unsupervised wav2vec pre-training [21] which we use in conjunction with SpeechStew training.

The key contributions of this paper are as follows:

1. We demonstrate that simply mixing multiple datasets and training a single large end-to-end speech recognition model, called SpeechStew, can yield strong empirical results.
2. We demonstrate the transfer learning capabilities of SpeechStew. When encountering a new dataset, one could simply fine-tune a generic pretrained model on unseen data to yield strong empirical results.

2. SpeechStew

In this section, we describe the model and training data setup of SpeechStew. We also describe our transfer learning setup for fine-tuning on new unseen tasks.

2.1. Model

In our implementation, SpeechStew uses the Conformer [29] RNN-T [30] architecture. We experiment with both the 100M parameter [29] and the 1B parameter configuration [5]. We find that wav2vec pre-training [15] is needed to train the 1B parameter model [5]. We apply the default hyperparameters from prior work [29, 5] including the learning rate schedule. We do not incorporate an external language model.

2.2. Multi-domain Training

We combine the following datasets without any form of re-weighting or resampling to construct the training set for SpeechStew:

1. AMI [31]. AMI is approximately 100 hours of meeting recordings.
2. Common Voice [32]. Common Voice is a crowd-sourced open licensed speech dataset. We use the version 5.1 (June 22 2020) snapshot with approximately 1500 hours. The data was collected at 48 KHz, and we resampled it to 16 KHz.
3. English Broadcast News (LDC97S44, LDC97T22, LDC98S71, LDC98T28). English Broadcast News is approximately 50 hours of television news.
4. LibriSpeech [33]. LibriSpeech is approximately 960 hours of speech from audiobooks.
5. Switchboard/Fisher (LDC2004T19, LDC2005T19, LDC2004S13, LDC2005S13, LDC97S62). Switchboard/Fisher is approximately 2000 hours of telephone conversations. The data was collected at 8 KHz, and we upsampled it to 16 KHz.
6. TED-LIUM v3 [34, 35]. TED-LIUM is approximately 450 hours of TED talks.
7. Wall Street Journal (LDC93S6B, LDC94S13B). WSJ is approximately 80 hours of clean speech.

2.3. Transfer Learning

We demonstrate the transfer learning capabilities of SpeechStew. Once we have a general purpose SpeechStew model (trained on the datasets mentioned in Section 2.2), we can fine-tune and adapt SpeechStew onto a new task. CHiME-6 [7] is a noisy low resource dataset set, which contains approximately 40 hours of distant microphone conversational speech recognition in everyday home environments. CHiME-6 is difficult for end-to-end speech recognition models to train directly due to over-fitting issues [8]. We fine-tune SpeechStew on CHiME-6 to demonstrate the transfer learning capabilities.

The transfer learning capabilities of SpeechStew are extremely practical. It implies we can train a general purpose model once, then fine-tune to specific low resource tasks. This can be done at a very low cost, since fine-tuning typically requires only a few thousand steps, compared to ≈100k steps needed to train a model from scratch.

3. Experiments

As mentioned in Section 2.2, we use AMI, Common Voice v5.1, English Broadcast News, LibriSpeech, Switchboard/Fisher, TED-LIUM v3, and Wall Street Journal to train our SpeechStew model. For AMI, we use both individual headset microphone (IHM) and single distant microphone (SDM1) data. We resampled the data to 16 KHz for Common Voice and Switchboard/Fisher (which were 48 KHz or 8 KHz respectively). We use 80 dimensional filter bank coefficients as our input features. We evaluate our model on several tasks: AMI (IHM and SDM1), Common Voice, LibriSpeech (clean and other), Switchboard (Switchboard and CallHome subsets of Hub5'00), Tedlium, and WSJ eval92. We score our results with Kaldi scripts [44]. For Common Voice, we evaluate with and without an extra step of text normalization by dropping punctuations to make our results comparable to [36]. We do not apply any other form of text normalization during training or evaluation.

We build single task mode baselines, where the models are trained only on their respective domains. We use the Conformer 100M architecture [29] for these baselines; we found the 1B model to overfit and perform dramatically worse. We perform model selection via the development sets per baseline task.

Our SpeechStew model uses the 100M parameter and 1B parameter Conformer architecture [29, 5]. We used the default experimental settings of these references to train the models. We train all our SpeechStew models for exactly 100k steps, without any model selection.

The 100M parameter model, referred to as "Conformer L" in [29], is trained for 100k steps using Adam optimization with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and a transformer learning rate schedule (section 5.3 of [25]) with 10k warm-up steps and peak learning rate 2e-3. The batch size is set to 8192. We find that using a large batch is important for getting good results. Residual dropout with rate 0.1 is applied, as is weight-noise [45] and L2-regularization with weight 1e-6. Exponential-moving-averaged model weights with decay rate 0.9999 are used for evaluating the model. We apply adaptive SpecAugment [4, 46] with two frequency masks of size parameter $F = 27$, and ten time masks with maximum time-mask ratio $p_S = 0.05$ to augment the input spectrogram.

For the 1B parameter model, "Conformer XXL" of [5], we follow this reference to apply LibriLight wav2vec 2.0 [15] pre-training to initialize the network. LibriLight [47] is approximately 60k hours of unlabelled data, in-domain with LibriSpeech. As in [5], we found the wav2vec pre-training to be critical to optimize these large neural networks. Following [15], we use non-quantized pre-training with sampling probability 0.065 and mask size 10. The wav2vec contrastive loss is optimized via Adafactor optimization [48] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$. We use a transformer learning rate schedule for pre-training with peak learning rate 2e-3 and 25k warm-up steps. Pre-training the encoder of the model for 400k steps, we carry out supervised training on SpeechStew for 100k steps with batch size 2048. The encoder and decoder are optimized separately with two Adafactor optimizers (with the same beta parameters as in pre-training) and two transformer learning rate schedules. Peak learning rates 3e-4/1e-3 and warm-up steps 5k/1.5k are used for the encoder/decoder schedules, respectively. The same SpecAugment policy as the 100M model is used to train this model, while exponential moving averages are not used.

Table 1 summarizes our results. We emphasize that the reported performance of SpeechStew is obtained without the use of an external language model.

SpeechStew 100M outperforms almost all prior work, including those using strong language models. On AMI-IHM and AMI-SDM1, SpeechStew obtains 9.0 WER and 21.7 WER respectively. On Common Voice, SpeechStew obtains 12.1 WER or 9.7 WER with punctuation normalization. On LibriSpeech, SpeechStew obtains 2.0 and 4.0 on the clean and other test splits. These results are worse than [5], where noisy student training [17] utilizing extra unlabelled data (LibriLight)

Task	AMI		Common Voice	LibriSpeech		Switchboard/Fisher		Tedlium	WSJ
	IHM	SDM1		clean	other	SWBD	CH		eval92
Prior Work (no LM)									
Single domain			16.9† [36]	1.5 [5]	2.7 [5]			7.5 [36]	9.3 [37]
Multi-domain	12.2‡ [38]	**21.2**‡ [38]	15.5† [36]	3.0 [36]	7.3 [36]	6.3 [36]	10.7 [36]	6.9 [36]	3.4 [36]
Prior Work (with LM)									
Single domain	17.5 [39]	36.4 [40]	13.6† [36]	**1.4** [5]	**2.6** [5]	4.9 [41]	9.5 [41]	5.6 [42]	2.9 [43]
Multi-domain			10.6† [36]	2.1 [36]	4.4 [36]	5.5 [36]	9.1 [36]	**5.2** [36]	2.0 [36]
Our Work (no LM)									
Single Domain Baseline (100M)	26.1	40.5	16.3 (13.8†)	2.1	4.4	5.6	9.7	7.6	28.2
SpeechStew (100M)	**9.0**	**21.7**	12.1 (9.7†)	2.0	4.0	**4.7**	**8.3**	**5.3**	**1.3**
SpeechStew (1B)	9.5	22.7	**10.8 (8.4†)**	**1.7**	**3.3**	4.8	10.6	5.7	**1.3**

Table 1: *Speech recognition word error rates (%) across multiple tasks. SpeechStew achieves SoTA or near SoTA across many tasks. Our SpeechStew 1B model uses wav2vec pre-training on LibriLight. SpeechStew does not use a separate language model.* †*We follow [36] and remove punctuations during evaluation.* ‡*Concurrent work [38].*

Model	**Dev**	**Eval**
Prior Work (with LM)		
Official HMM Baseline [7]	51.8	51.3
HMM [9]	**36.9**	**38.6**
RNN-T [8]	49.0	
Our Work (no LM)		
Zero-Shot (never seen CHiME-6)		
SpeechStew (100M)	54.9	57.2
SpeechStew (1B)	39.2	53.7
Fine-tuned with CHiME-6		
Baseline (100M)	70.0	66.7
SpeechStew + Fine-tune (100M)	33.1	40.6
SpeechStew + Fine-tune (1B)	**31.9**	**38.9**

Table 2: *We apply transfer learning and fine-tune SpeechStew on CHiME-6.*

has been applied. On Switchboard/Fisher, SpeechStew obtains 4.7 WER on the Switchboard split, and 8.3 WER on Call-Home split. On TED-LIUM v3, SpeechStew obtains 5.3 WER, slightly behind the 5.2 WER of [36], which has been obtained with the use of a language model. On WSJ, SpeechStew obtains 1.3 WER. To the best of our knowledge, our SpeechStew model outperforms all prior work on AMI, Common Voice, Switchboard and WSJ. Overall, we find SpeechStew to achieve SoTA or near-SoTA across a wide variety of tasks.

SpeechStew 1B with LibriLight wav2vec pre-training [15] exhibits improved performance on Common Voice and LibriSpeech compared to the 100M model, while the other tasks suffer from a small relative degradation. One possible explanation for this phenomenon is that the LibriLight pre-training data used to initialize the network is in-domain with the LibriSpeech task, suggesting that the unsupervised pre-training data may bias the network toward a certain domain. We, however, also find that the 1B model exhibits better transfer learning capabilities in the next section.

3.1. Transfer Learning and CHiME-6

CHiME-6 [7] is a low resource noisy speech dataset. It is especially challenging due to the small size and noisy audio conditions. We perform front end enhancement following the official recipe [7]. We use BeamformIt to enhance our front end during training. We use guided source separation [49] with 12 channels to enhance our front end for evaluation.

We fine-tune our 100M and 1B SpeechStew models with the CHiME-6 data, and compare these results against baselines obtained by training a 100M parameter Conformer model and a 1B-parameter Conformer model (with LibriLight-only pre-training) with CHiME-6. Table 2 summarizes our results. We have not recorded results for the 1B-parameter baseline model, as we have failed to train the model to exhibit non-trivial performance. We have adjusted the dropout rate, the weight-decay parameter, augmentation parameters, learning rate and warm-up steps with respect to the default values from [29] and [5] to optimize the dev-set WER.

For the 100M models, we apply a universal dropout rate for the input, residual connections for the feed-forward, convolutional and attention units. For the baseline model, a universal dropout rate of 0.5 is applied, with L2-regularizer weight 1e-4. We only use two time masks with parameters $T = 80$ and maximum time-mask ratio $p_S = 0.25$ for augmenting the input. The default values for the learning rate schedule (2e-3 peak learning rate with 10k warm-up steps) are used. When fine-tuning from SpeechStew, a universal dropout rate of 0.4 and L2-regularizer weight 1e-6 is used. The SpecAugment parameters are set to be equivalent to the baseline, while the peak learning rate and warm-up steps for the learning rate schedules are set to 1e-4 and 2k steps, respectively. The best dev-set performance when fine-tuning from SpeechStew is achieved at 4k steps, compared to 60k steps for the baseline.

When fine-tuning the 1B SpeechStew model, L2-regularization is turned off. SpecAugment is applied with three frequency masks with $F = 10$ and eight time masks with $T = 96$ and $p_S = 0.08$. Both the encoder and decoder learning rate schedule parameters are set to have peak learning rate 2.5e-4 and 8k warm-up steps. Best dev-set performance is achieved upon 6k steps of training.

The official CHiME-6 HMM baseline [7] achieves 51.8 and 51.3 WER on the dev and eval set respectively. A strong HMM model with a strong language model [9] achieves 36.9 and 38.6 WER on the dev and eval set respectively. The previous best end-to-end model [8] achieves 49.0 on the dev set. The SpeechStew 1B parameter model in the zero-shot setting (having never seen any CHiME-6 data before), achieves 39.2 and 53.7 WER on the dev and eval set respectively. Further fine-tuning our

SpeechStew model on the CHiME-6 data results in 31.9 and 38.9 WER on the dev and eval set respectively. We note that our results, obtained without using any external language model, compares to prior work that have utilized strong language models. This demonstrates that the SpeechStew model already possesses a significant amount of knowledge that is useful for the CHiME-6 task.

We make two observations regarding the effectiveness of SpeechStew on ChiME-6 in particular. The first is that the amount of training time spent on the low-resource CHiME-6 data is greatly reduced. This is a natural product of the finetuning process. However, SpeechStew's construction allows us to skip to that stage of the training process, thus reducing the risk of overfitting. Secondly, SpeechStew appears to be robust enough to handle noisy data. SpeechStew spends more time learning the salient features from a larger pool of less noisy data, and less time working with the more noisy CHiME-6 data.

4. Related Work

Mixing multiple datasets together to train one neural network is not new. Kaldi's `multi_en` recipe [12] has a very similar setup to ours. Narayanan et al. [13] mixed several proprietary commercial datasets, totally more than 160k hours of training data to train one neural network as well. Likhomanenko et al. [36] also mixed several datasets together to train end-to-end speech recognition models. One of the key difference between our work and prior work is the model size, we scale to much larger models.

Neural networks often benefit from scaling up the number of parameters in practice [27]. This has been observed in language modelling [14, 50, 26], computer vision [24, 50, 51], and speech recognition [52, 5]. Similarly, SpeechStew leverages large 100M and 1B parameter models to yield strong empirical results. However, our model sizes still pale in comparison to those found in the language modelling literature, for example 175B parameters in GPT-3 [26].

Transfer learning is a popular technique found in computer vision [53, 19, 22] and natural language processing [14, 26]. A general purpose model, which is trained either on a large corpus of unlabeled data or labeled data, is then finetuned on a low resource task. SpeechStew also adopts the pretrain, finetune setup for transfer learning. We train SpeechStew on a large labeled corpus, and finetune on CHiME-6. There has also been prior work on pretraining on unlabelled audio, and finetuning on supervised data [21, 15]. Our work is focused on supervised pretraining and supervised finetuning. However, we still leverage unsupervised wav2vec pretraining in our 1B parameter models. We believe these techniques may be additive, especially as one scales to very large models. Finally, the concurrent work of [38] also applied supervised pretraining and finetuning; they pre-trained on 75k hours of proprietary labelled data, and finetuned on AMI to achieve very strong emprical results.

5. Discussion

Deep learning models have made significant progress from two simple principles:

1. Train on more data [23, 26].
2. Train larger and deeper neural networks [24, 14].

Supervised data is expensive to acquire. SpeechStew tackles this problem by simply mixing all publicly available speech recognition data. Our approach leverages on currently available resources, labelled and unlabelled. We hope our work will encourage future research to leverage on all training data available, as opposed to training on only task specific datasets.

Training large models is expensive, and impractical to do frequently (especially when one regularly encounters new tasks or new data). Our work on transfer learning demonstrates that one can simply finetune a pretrained model for only a few thousand gradient steps and achieve strong results. This is inexpensive and very practical. We hope our work will encourage further research to leverage on transfer learning in speech recognition.

6. Acknowledgements

We give thanks to Chung-Cheng Chiu, David Fleet, Naoyuki Kanda, Bo Li, and Samy Bengio for their discussions and review of the manuscript.

7. References

[1] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *ICML*, 2014.

[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.

[3] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence Modelling via Imputation and Dynamic Programming," in *ICML*, 2020.

[4] D. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. Cubuk, and Q. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *INTERSPEECH*, 2019.

[5] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition," in *arXiv:2010.10504*, 2020.

[6] C.-C. Chiu, T. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art Speech Recognition With Sequence-to-Sequence Models," in *ICASSP*, 2018.

[7] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, and N. Ryant, "CHiME-6 Challenge:Tackling Multispeaker Speech Recognition for Unsegmented Recordings," in *arXiv*, 2020.

[8] A. Andrusenko, A. Laptev, and I. Medennikov, "Towards a Competitive End-to-End Speech Recognition for CHiME-6 Dinner Party Transcription," in *arXiv*, 2020.

[9] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya1, I. Soroki, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, "The STC System for the CHiME-6 Challenge," in *CHiME Workshop*, 2020.

[10] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey, A. Rastrowand, R. C. Rose, and S. Thomas, "Multilingual Acoustic Modeling for Speech Recognition based on Subspace Gaussian Mixture Models," in *ICASSP*, 2010.

[11] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual Acoustic Models Using Distributed Deep Neural Networks," in *ICASSP*, 2013.

[12] https://github.com/kaldi-asr/kaldi/commits/master/egs/multi_en.

[13] A. Narayanan, A. Misra, K. C. Sim, G. Pundak, A. Tripathi, M. Elfeky, P. Haghani, T. Strohman, and M. Bacchiani, "Toward domain-invariant speech recognition via large scale training," in *SLT*, 2018.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *NAACL*, 2019.

[15] A. M. M. A. Alexei Baevski, Henry Zhou, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *arXiv:2006.11477*, 2020.

[16] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end ASR: from Supervised to Semi-Supervised Learning with Modern Architectures," in *ICML: Workshop on Self-supervision in Audio and Speech*, 2020.

[17] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved Noisy Student Training for Automatic Speech Recognition," in *INTERSPEECH*, 2020.

[18] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," in *ICML*, 2014.

[19] S. Kornblith, J. Shlens, and Q. Le, "Do Better ImageNet Models Transfer Better?" in *CVPR*, 2019.

[20] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are All You Need: End-to-End Multilingual Speech Recognition and Synthesis with Bytes," in *ICASSP*, 2019.

[21] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," in *arXiv*, 2019.

[22] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big Transfer (BiT): General Visual Representation Learning," in *arXiv*, 2019.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NIPS*, 2017.

[26] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *NeurIPS*, 2020.

[27] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep Double Descent: Where Bigger Models and More Data Hurt," in *ICLR*, 2020.

[28] Y. Zhang, W. Chan, and N. Jaitly, "Very Deep Convolutional Networks for End-to-End Speech Recognition," in *ICASSP*, 2017.

[29] 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 *INTERSPEECH*, 2020.

[30] A. Graves, "Sequence Transduction with Recurrent Neural Networks," in *ICML Representation Learning Workshop*, 2012.

[31] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The AMI Meeting Corpus: A Pre-Announcement," in *International Workshop on Machine Learning for Multimodal Interaction*, 2005.

[32] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common Voice: A Massively-Multilingual Speech Corpus," in *LREC*, 2020.

[33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[34] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an Automatic Speech Recognition dedicated corpus," in *LREC*, 2012.

[35] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation," in *SPECOM*, 2018.

[36] T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve, "Rethinking Evaluation in ASR: Are Our Models Robust Enough?" *arXiv preprint arXiv:2010.11745*, 2020.

[37] S. Sabour, W. Chan, and M. Norouzi, "Optimal Completion Distillation for Sequence Learning," in *ICLR*, 2019.

[38] N. Kanda, G. Ye, Y. Wu, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Large-Scale Pre-Training of End-to-End Multi-Talker ASR for Meeting Transcription with Single Distant Microphone," in *arXiv*, 2021.

[39] G. Sun, C. Zhang, and P. C. Woodland, "Transformer Language Models with LSTM-based Cross-utterance Information Representation," in *arXiv*, 2021.

[40] N. Kanda, Y. Fujita, S. Horiguchi, R. Ikeshita, K. Nagamatsu, and S. Watanabe, "Acoustic Modeling for Distant Multi-talker Speech Recognition with Single- and Multi-channel Branches," in *ICASSP*, 2019.

[41] W. Wang, G. Wang, A. Bhatnagar, Y. Zhou, C. Xiong, and R. Socher, "An investigation of phone-based subword units for end-to-end speech recognition," in *INTERSPEECH*, 2020.

[42] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The RWTH ASR System for TED-LIUM Release 2: Improving Hybrid HMM with SpecAugment," in *ICASSP*, 2020.

[43] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *INTERSPEECH*, 2018.

[44] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *ASRU*, 2011.

[45] A. Graves, "Practical Variational Inference for Neural Networks," in *NIPS*, 2011.

[46] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Specaugment on large scale datasets," in *ICASSP*, 2020.

[47] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-Light: A Benchmark for ASR with Limited or No Supervision," in *arXiv*, 2019.

[48] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *ICML*, 2018.

[49] C. Boeddecker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-End Processing for the CHiME-5 Dinner Party Scenario," in *CHiME Workshop*, 2018.

[50] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, and D. A. Jeffrey Wu, "Scaling Laws for Neural Language Models," in *arXiv*, 2020.

[51] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big Self-Supervised Models are Strong Semi-Supervised Learners," in *NeurIPS*, 2020.

[52] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. Le, and A. Ng, "Large Scale Distributed Deep Networks," in *NIPS*, 2012.

[53] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *International Conference on Multimedia*, 2014.