Scaling End-to-End Models for Large-Scale Multilingual ASR

Bo Li, Ruoming Pang, Tara N. Sainath, Anmol Gulati, Yu Zhang,
James Qin, Parisa Haghani, W. Ronny Huang, Min Ma

Google LLC, USA

{boboli,rpang,tsainath}@google.com

Abstract

Building ASR models across many language families is a challenging multi-task learning problem due to large language variations and heavily unbalanced data. Existing work has shown positive transfer from high resource to low resource languages. However, degradations on high resource languages are commonly observed due to interference from the heterogeneous multilingual data and reduction in per-language capacity. We conduct a capacity study on a 15-language task, with the amount of data per language varying from 7.7K to 54.7K hours. We adopt GShard [1] to efficiently scale up to 10B parameters. Empirically, we find that (1) scaling the number of model parameters is an effective way to solve the capacity bottleneck - our 500M-param model is already better than monolingual baselines and scaling it to 1B and 10B brought further quality gains; (2) larger models are not only more data efficient, but also more efficient in terms of training cost as measured in TPU days - the 1B-param model reaches the same accuracy at 34% of training time as the 500M-param model; (3) given a fixed capacity budget, adding depth usually works better than width and large encoders tend to do better than large decoders.

Index Terms: large-scale, multilingual speech recognition

1. Introduction

Large data sets and giant models, together with advances in deep learning algorithms and hardware, enable researchers to push the limit of many machine learning tasks [1–3] including speech [4]. It brings renewed interests in building universal automatic speech recognition (ASR) systems that can recognize speech from any language. Traditionally, a predefined language agnostic representation is required for building multilingual models such as a global phoneme set like International Phonetic Alphabet (IPA) [5], Speech Assessment Methods Phonetic Alphabet (SAMPA) [6] and Worldbet [7] or a universal speech representation like articulations [8], which all require expert knowledge. In the past decade, deep neural networks have been widely adopted in the speech community [9], especially the recently developed end-to-end (E2E) models that merges the modeling of acoustics, lexicon and language into a single neural network [10–14]. It largely simplifies the building of multilingual models by learning shared representation directly from data. This also speeds up the sharing of techniques between ASR and other machine learning fields such as neural machine translation [15]. Experiments on less than 10 languages have shown promising capabilities of such E2E models in modeling dialects of a particular language [16], languages from the same family [17] and languages from different families [18–21]. Recently, massively multilingual experiments using more than 50 languages [19–21] also show comparable or better performance compared to monolingual systems.

A major focus of multilingual ASR has been improving performance on low resource languages, which benefit from the data pooling of similar languages, the cross language joint optimization and the consequence positive transfer from higher resource languages [19–23]. However, high resource languages where sufficient monolingual data exists, suffer from interference and constrained capacity [19,24], and often see a degradation in performance. Improving performance across the board on both high and low resource languages is an under-studied and challenging task.

From a machine learning perspective, a statistical model's generalization capability builds on inductive bias of the learning algorithms [25]. The underlying inductive bias for multilingual systems is that the learning signal from one language should benefit the quality of other languages [26]. Under this assumption, the model will generalize better with an increasing number of languages due to the additional information brought by the new languages. This positive transfer is best observed for low resource languages. However, as we increase the number of languages, the modeling task become more challenging due to the large language variations and heavy data imbalance [24]. With a fixed model capacity which is loosely measured in terms of the number of free parameters in neural networks, the positive/negative transfer boundary becomes salient, and high resource languages start to regress due to task interference and reduction in per-task capacity. In [27], a simple and scalable optimization procedure, namely Gradient Vaccine, is developed to address the gradient interference problem from different tasks. Distilling knowledge from single task models to the multi-task model [28] has also been found to address the interfere problem.

In this paper, we investigate the high resource language performance degradation problem of multilingual models from a capacity perspective [1, 15]. Prior work explored as many as 50 [19] to 100 [21] languages. However, the scale of the dataset is very limited. The largest language used in [19] has just over 1K hours of speech. In our study, the amount of data per language ranges from 7.7K to 54.7K hours, which leads to high quality monolingual baselines, posing a challenge to build a single multilingual model that can outperform them. We present a capacity solution with thorough empirical studies demonstrating how it is devised. Unlike [19], no details were provided on how their best 1B model was developed. We adopt the GShard [1] technique to efficiently scale our model up to 10B which shows further word error rate (WER) reductions though relatively small. With the increase of model capacity, we manage to recover the performance of monolingual models on all the high resource languages. We ablate the various factors in increasing the model capacity and find that depth generally does better than width and encoder capacity correlates well with recognition performance. We observe that with a fixed capacity, how to feed the language information becomes less important. Moreover, large models are more sample [29] and cost efficient, requiring fewer training iterations and less TPU time to reach similar performance.

2. Multilingual E2E Models

2.1. Model Architecture

Our multilingual ASR system is an attention-based encoder-decoder model. For encoder, we use full-context Conformer layers [30], consisting of an input projection layer, a relative position embedding layer followed by a stack of Conformer layers which are organized into three blocks. The first Conformer block consists of 4 Conformer layers followed by a time stacking layer that concatenates the current output with one frame on its left. This doubles the output dim while achieving a 2X time reduction. The second block consists of a single Conformer layer and a projection layer which halves the feature dim and brings it back to the same dimension as the other layers. The remaining Conformer layers make up the third block. Similar to [31], we use the existing convolution module to provide relative positional information and group normalization to address variations across languages in each Conformer layer.

We experiment with two types of decoders. Firstly, a unidirectional Long Short-Term Memory (LSTM) [32] based decoder is used together with a 4-head additive content-based attention to form a Listen, Attend and Spell (LAS) model [33]. Secondly, we also adopt a Transformer decoder with masked self attention and cross attention to the encoder outputs [13, 34].

The output vocabulary of our multilingual ASR is a unified grapheme set with 3328 tokens; among those 3,315 tokens appear at least 1000 times in the training data and the remaining are special tokens like "<s>", "</s>" and padded placeholders. The majority of the graphemes (3,055) come from Chinese; even with that, Chinese is the only language that has OOV grapheme problem due to the selection threshold and the coverage of training data. We feed language information via a one-hot embedding vector into the encoder as either an additional input [16] or the switch for language adapters [17, 28]. We simply pool all the data together and sample each batch according to the natural distribution. The whole encoder-decoder network is jointly optimized to minimize the cross-entropy loss between the output of the network and the ground truth transcripts.

2.2. Scaling Up Model Capacity

There are multiple ways to scale up an encoder-decoded based multilingual mode. In this work, we empirically study the effect of the following factors: **(1)** width *vs.* depth; **(2)** encoder *vs.* decoder; **(3)** language dependent capacity *vs.* language independent capacity; **(4)** architecture *vs.* capacity. Strictly speaking, model capacity is not equivalent to the number of parameters *i.e.* the model size. For models with language dependent components built in such as adapter models [28], the inference capacity is smaller than training as during inference only the shared parameters and those corresponding to a specific language are active. To simplify the discussion, we look at the training model capacity and use model size and capacity interchangeably.

Scaling up models comes with various practical challenges: the model parallelism support, the computation cost, and the infrastructure support, etc. Recently, the GShard annotation API has been developed for parallel execution [1] and is released in Lingvo [35] which makes building giant models simpler. Additionally, a new compiler infrastructure, namely the Single Program Multiple Data (SPMD) partitioner , is developed, which makes the compilation time near constant regardless of the number of partitions [1]. This allows us to more efficiently scale to thousands of partitions. We hence adopt these advances to scale up our multilingual models to 1B parameters and beyond.

Table 1: *Per language training data statistics. Utterance counts are in millions (M) and durations are in thousand (K) hours.*

Language	Family	Counts(M)	Hours(K)
English (US)	Germanic	35.3	54.7
English (IN)		18.2	27.6
Spanish (US)	Italic	31.8	48.2
Portuguese (BR)		18.1	33.4
Spanish (ES)		16.4	24.0
Arabic (GULF)	Arabic	7.8	12.0
Arabic (EG)		7.7	12.0
Hindi (IN)	Indo-Iranian	20.1	32.8
Marathi (IN)		11.6	16.9
Bangladesh (BD)		8.7	16.7
Chinese (TW)	Chinese	17.5	23.2
Russian (RU)	Balto-Slavic	14.9	23.0
Turkish (TR)	Turkic	15.9	22.6
Hungary (HU)	Uralic	6.7	10.1
Malay (MY)	Austronesian	4.7	7.7
Total		235.4	364.9

3. Experimental Details

3.1. Data

Experiments are conducted on a dataset of 15 languages from 9 distinct language families. There are totally 235.4M utterances which correspond to 364.9K hours of speech data from Google's Voice Search traffic. This is more than 20 times of the data used in [19]. To the best of our knowledge, this is the first work looking at building multilingual ASRs at such a large scale. The data is annonymized and human transcribed. Per language data statistics are listed in Table 1. The number of utterances for each language ranges from 4.7M to 35.3M, roughly corresponding to 7.7K to 54.7K hours of speech data. The unbalanced data distribution poses a modeling challenge. Unlike other existing multilingual work, we focus on investigating the interference problem between high resource languages. The smallest language in our setup has around 7.7K hours of training data, which is about 7 times of the highest resource language used in [19]. This large scale dataset again bring in training efficiency challenges. The test set for each language contains around 3~19K utterances sampled from Google's Voice Search traffic with no overlapping from the training set. Similarly, they are anonoymized and hand-transcribed for evaluation purpose.

3.2. Training

We use 80D log Mel features that are computed using 32ms windows with a 10ms hop. Features from 3 contiguous frames are stacked and subsampled to form a 240D input representation with 30ms frame rate. A 16D one-hot language vector is fed into the encoder as an additional input. SpecAugment [36] is used to improve models' robustness against noise. Specifically, two frequency masks with a maximum length of 27 and two time masks with a maximum length of 50 are used.

All the models are trained in Tensorflow [37] using the Lingvo [35] toolkit on Google's Tensor Processing Units (TPU) V3 with a global batch size of 4,096 utterances. Models are trained with 512 TPU cores except for 10B models which use

1024 TPU cores. This is mainly due to the 16G per core high bandwidth memory (HBM) limit. Models are optimized using synchronized stochastic gradient descent. For LSTM-decoder models, we use the Adam optimizer [38] with parameters β_1=0.9 and β_2=0.999; for Transformer-decoder models, Adafactor [39] with parameters β_1=0.9 and β_2=0.99 is used. A transformer learning rate schedule [34] with peak learning rate 3e-4 and 10K warm-up steps is used.

4. Results and Discussions

In this section we present our study of building high quality multilingual models on large scale dataset. For simplicity, we use the average WER for comparisons and only report the per language performance at the end.

4.1. Monolingual Baselines

Conformer has been shown to perform the best on many English tasks [30, 31]. We hence adopt it for our monolingual baselines, specifically we use the same Conformer encoder and LSTM decoder architecture as [31] but in a full context setup. The encoder consists of 17 layers of Conformer blocks with 12 layers in the last block (*c.f.* Section 2.1). Each Conformer layer has a model dimension of 512. 8-head attention is used in the self-attention layer and the Convolution kernel size used is 15. The decoder is an LSTM based LAS decoder, consisting of 2 layers of 640D LSTM with 2048 hidden units. 4-head content-based additive attention is used in the LAS attention module. Each monolingual model has around 140M parameters and is trained to predict language dependent graphemes only. The average WER is 9.29% and the per language breakdown is depicted in Figure 1(c). Across languages, WER ranges from English (US)'s 4.6% to Marathi (IN)'s 20.2%. Languages with more data tend to have lower WERs.

4.2. Multilingual Encoder Architecture

To justify the effectiveness of Conformer encoders for multilingual modeling, we compare three encoders with the same LSTM-based decoder: **(1) LSTM encoder** with 8 layers, 2048D hidden units and 640D output units; **(2) ContextNet encoder** with 24 layers, 640D hidden units and channel scale of 2 [40]; and **(3) Conformer encoder** with 17 layers, 512D hidden dimension the same as the monolingual baselines. Language adapters are inserted between each encoder layers. The specific configurations of these three encoders are chosen such that the total number of model parameters are roughly the same, which is around 220M. The increase of model size compared to monolingual models comes from the additional language adapters and the increase of output vocabulary size. The average WER of these three models are 11.86%, 10.77% and 9.43% respectively. This clearly demonstrates the effectiveness of Conformer for multilingual ASR. Comparing to monolingual models, even though this baseline multilingual Conformer still lags behind in quality, it does reasonably well in recognizing all the 15 languages. It converges in around 1.2M training steps which roughly corresponds to 21 epochs, while the monolingual models normally train up to 50 epochs.

To understand the effect of the language adapters, we conduct the following ablation studies. For quick experimentation, we compare models at 200K steps (roughly 3.5 epochs), which we find sufficient for model selection.

The necessity of language dependent parameters. The use of language adapters brings in both language dependent model parameters and a small increase in model size. To understand which helps more, we train a single adapter model that forces all the languages to share the same adapter transformation. In this way we can isolate the model size increase from the adapter model. This model achieves 10.86% average WER *vs.* the baseline language dependent adapter model's 10.38% @200K steps. This suggests feeding in language information and learning language dependent parameters are important.

The necessity of language dependent transforms. A simpler way of incorporating language information is to append the one-hot language vector to the input features. It effectively adds language dependent biases; while adapters bring in additional language dependent transformations. At 200K steps, the bias-only model achieves an average WER of 10.93% which is worse than the adapter model. However, one thing to note is it has less number of parameters (146M *vs.* 220M). We further increase the bias-only model to the same 220M, which yields an average WER of 10.37% @200K steps similar to the adapter model. Though they have the same amount of parameters, the bias-only model has slightly higher inference cost than the adapter model whose adapter components are partially activated depending on the language. But for simplicity, we will iterate base on the bias-only Conformer encoder model.

4.3. Multilingual Decoder Architecture

Besides using a single shared decoder, multi-head models [19] that use different decoders for different languages/families can be used to add per task capacity. Similarly to [19], we use per language family decoders. Totally 5 families are used: Germanic, Italic, Arabic, Indo-Iranian and others. For comparison, we ensure the single decoder and the multi-decoder model have the same number of parameters: **(1) single-decoder** has 6 layers of 768D LSTM with 3074 hidden units and **(2) multi-decoder** has 5 decoders and each have 2 layers of 640D LSTM with 2048 hidden units. Both models have 354M parameters. At 200K steps, single-decoder yields of an average WER of 10.13% compared to the multi-decoder's 10.28%. This suggests given the same model size, it's more beneficial to use a single decoder which encourages more cross language/family sharing.

To push the performance of our multilingual model, we further increase the 354M model to 500M by increasing the encoder's width from 512D to 640D and depth from 17 to 22 layers. It reaches an average WER of 9.63% @200K and converges to 9.13% @1.1M steps, outperforming the monolingual models. However, its training speed is less than 1/3 of the baseline 220M model due to the error back propagation through time for RNN models. This makes it unfavourable for further scaling up.

Transformer decoder [34] instead does not have the time recurrence constraint and has much high parallelism in training. With the same encoder architecture, we build a Transformer decoder model with totally 500M parameters, which leads to 12 Transformer layers with 768D model dim, 3072D hidden dim and 8 attention heads. It converges to a slightly higher WER of 9.26% but with a training speed similar to the 220M baseline. We hence use Transformer decoders for following studies.

4.4. Scaling up with GShard

In this experiment, we want to find the best way to further scale up the Conformer encoder and Transformer decoder model to 1B parameters. The set of experiments are listed in Table 2. Comparing E1 *vs.* E2 and E5 *vs.* E6, for both encoder and decoder, deeper model has better WER than wider models. However, deeper models are slow to train. Comparing E1-E4 with
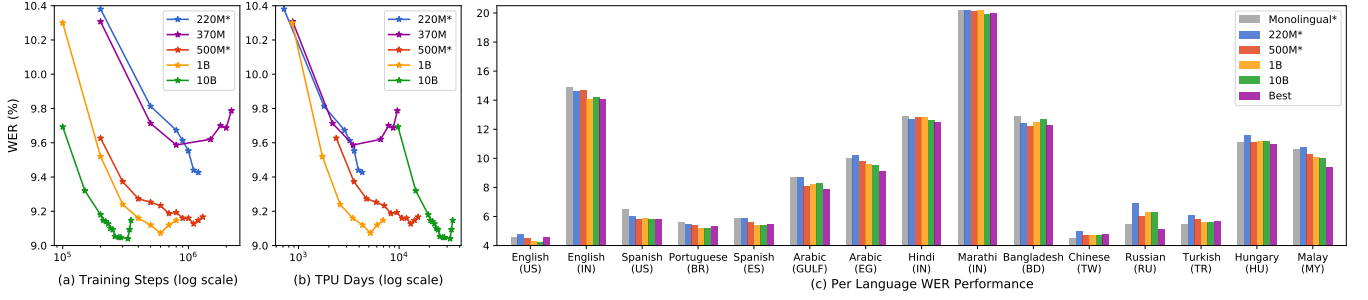
Figure 1: *WER performance (%) vs.* **(a)** *training steps,* **(b)** *TPU days and* **(c)** *language. Systems with * use LSTM decoders.*

Table 2: *1B-param model comparisons (***L***: number of layers;* **W***: model dim;* **Loss***: training negative log perplexity, lower is better;* **Speed***: number of training examples per second). B0 is the baseline model and "-" indicates no change from B0.*

Exp.	Encoder		Decoder		Training		WER↓
	L	W	L	W	Loss↓	Speed↑	(@200K)
B0	17	768	12	768	0.158	5530	10.36
E1	61	-	-	-	0.155	2352	10.13
E2	-	1408	-	-	0.150	**3419**	10.17
E3	33	1024	-	-	0.149	2975	**10.05**
E4	26	1152	-	-	0.151	3198	10.23
E5	-	-	76	-	**0.143**	2111	10.15
E6	-	-	-	1920	0.149	3170	10.48
E7	-	-	22	1408	0.147	3204	10.37
E8	22	1024	18	1152	0.147	2870	10.08

E5-E7, adding capacity to encoder does slightly better than decoder in terms of WER, however, large decoder tends to have better training loss. This might suggest the decoder's modeling task of the current speech training data is relatively simpler than the encoder's modeling task and larger decoder shows signs of over fitting. E4 that equally splits the additional capacity to width and depth [29] does not work well on this task instead with more capacity allocated to depth (i.e. E3) we obtained the best WER. Lastly, E8, which first equally splits the capacity between encoder and decoder and then allocates more to depth, performs similarly to E3.

E3 converges around 600K steps, roughly 10 epochs, halving the number of training iterations needed for the smaller models though each step runs longer. It is more data efficient. More importantly, it achieves an average WER of 9.07%.

4.5. Towards 10B-Param Model

Based on the previous experiments, we further scale up the model size towards 10B parameters by focusing more capacity on the encoder and depth. Specifically, we increased the E3 encoder depth from 33 to 86 and width from 1024 to 2048 and kept the decoder the same. It converges to 9.04% at 330K steps (∼6 epochs). Although the WER reduction comparing to the 1B model is marginal, the larger capacity leaves more room for scaling up from 15 languages to more in future.

Besides the performance gains, we find that large models tend to be more data and training cost efficient, *i.e.* they can reach the same level of performance with fewer optimization steps (Figure 1(a)) and less TPU days (Figure 1(b)). This is similar to the observation in [29]. We did not scale beyond 10B

mainly due to the slow training speed with current hardware. As shown in Figure 1(a), 10B model has better sample efficiency than 1B, *i.e.* less training epochs are needed to reach the same WER, but the longer TPU time required per step makes it impractical for now. Sparse models [1] have been found to scale up more efficiently, which will be explored in future work.

4.6. Human-in-the-loop Data Balancing

For simplicity, we only compared the average WER across models. To understand how they do on each language, we plot the breakdown in Figure 1(c) for the monolingual, 220M, 500M, 1B and 10B at convergence. Larger models win over monolingual models on most languages; however, there are languages they lag behind, especially on Russian (RU). We suspect this is because of the unbalanced data distribution. The amount of data per language mainly depends on the data collection project and has less consideration of the language complexity itself. To validate this, we take the 1B model and increase the mixing ratio for Russian (RU) to 0.4 and all the others with the same weights to continue training for another 130K steps. This reduces the WER on Russian (RU) from 6.3% to 5.1% which wins over the monolingual's 5.5%. As we still maintain a small weight for others, no clear degradation is observed and the average WER is further reduced from 9.07% to 8.87%. This suggests it is beneficial to find a better data mixing ratio or a schedule of mixing ratios for multilingual models, which will be explored in future.

5. Conclusions

In this work, we investigate the problem of building multilingual end-to-end ASR models on high resource languages with large scale datasets, where language interference becomes more prominent. We address this problem by scaling up model capacities and empirically show that we can build models up to 10B parameters. With larger models, we have observed consistent performance gains. Moreover, the larger models are more sample and training cost efficient, *i.e.* requiring less training optimization steps and TPU time, though each training step of giant models runs longer. With increased capacities, we can build a single multilingual model that outperforms the monolingual models on high resource languages on a large scale multilingual dataset. We do see on some languages the multilingual model is still lagging behind. Empirical evidence suggests it is a data balancing problem, which will be investigated in future.

6. Acknowledgements

7. References

[1] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[2] 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.

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language Models are Few-Shot Learners," *arXiv preprint arXiv:2005.14165*, 2020.

[4] 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," *arXiv preprint arXiv:2010.10504*, 2020.

[5] I. P. Association, I. P. A. Staff *et al.*, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet.* Cambridge University Press, 1999.

[6] J. C. Wells, "Computer-coding the IPA: a proposed extension of SAMPA," *Revised draft*, vol. 4, no. 28, p. 1995, 1995.

[7] J. L. Hieronymus, "ASCII phonetic symbols for the world's languages: Worldbet," *Journal of the International Phonetic Association*, vol. 23, p. 72, 1993.

[8] J. Frankel and S. King, "ASR-articulatory speech recognition," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[10] J. Li, Y. Wu, Y. Gaur *et al.*, "On the Comparison of Popular End-to-End Models for Large Scale Speech Recognition," *arXiv preprint arXiv:2005.14327*, 2020.

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

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

[13] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 7829–7833.

[14] G. Saon, Z. Tueske, D. Bolanos, and B. Kingsbury, "Advancing rnn transducer technology for speech recognition," *arXiv preprint arXiv:2103.09935*, 2021.

[15] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M. X. Chen, Y. Cao, G. Foster, C. Cherry *et al.*, "Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges," *arXiv preprint arXiv:1907.05019*, 2019.

[16] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-Dialect Speech Recognition With A Single Sequence-To-Sequence Model," in *Proc. ICASSP*, 2018, pp. 4749–4753.

[17] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model," in *Proc. Interspeech*, 2019.

[18] 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 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 5621–5625.

[19] V. Pratap, A. Sriram, P. Tomasello *et al.*, "Massively multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters," in *Proc. Interspeech*, 2020.

[20] W. Hou, Y. Dong, B. Zhuang, L. Yang, J. Shi, and T. Shinozaki, "Large-Scale End-to-End Multilingual Speech Recognition and Language Identification with Multi-Task Learning," in *Proc. Interspeech*, 2020.

[21] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, "Massively multilingual adversarial speech recognition," *arXiv preprint arXiv:1904.02210*, 2019.

[22] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," *arXiv preprint arXiv:1806.05059*, 2018.

[23] E. Chuangsuwanich, "Multilingual techniques for low resource automatic speech recognition," Massachusetts Institute of Technology Cambridge United States, Tech. Rep., 2016.

[24] X. Wang, Y. Tsvetkov, and G. Neubig, "Balancing training for multilingual neural machine translation," *arXiv preprint arXiv:2004.06748*, 2020.

[25] T. M. Mitchell, *The need for biases in learning generalizations.* Department of Computer Science, Laboratory for Computer Science Research . . . , 1980.

[26] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[27] Z. Wang, Y. Tsvetkov, O. Firat, and Y. Cao, "Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models," *arXiv preprint arXiv:2010.05874*, 2020.

[28] W.-H. Li and H. Bilen, "Knowledge distillation for multi-task learning," in *European Conference on Computer Vision.* Springer, 2020, pp. 163–176.

[29] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[30] A. Gulati, J. Qin, C.-C. Chiu *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020.

[31] B. Li, A. Gulati, J. Yu *et al.*, "A Better and Faster End-to-End Model for Streaming ASR," in *Proc. ICASSP*, 2021.

[32] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in *Proc. Interspeech*, 2014.

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

[34] 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. [Online]. Available: http://arxiv.org/abs/1706.03762

[35] J. Shen, P. Nguyen, Y. Wu *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *arXiv preprint arXiv:2005.08100*, 2019.

[36] 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," *Interspeech 2019*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

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

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[39] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *International Conference on Machine Learning.* PMLR, 2018, pp. 4596–4604.

[40] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," in *Proc. Interspeech*, 2020.