MASSIVELY MULTILINGUAL ASR: A LIFELONG LEARNING SOLUTION

Bo Li, Ruoming Pang[*], *Yu Zhang, Tara N. Sainath, Trevor Strohman,*
Parisa Haghani, Yun Zhu, Brian Farris, Neeraj Gaur and Manasa Prasad

Google LLC, USA

{boboli,rpang,ngyuzh,tsainath}@google.com

ABSTRACT

The development of end-to-end models has largely sped up the research in massively multilingual automatic speech recognition (MMASR). Previous research has demonstrated the feasibility to build high quality MMASR models. In this work, we study the impact of adding more languages and propose a lifelong learning approach to build high quality MMASR systems. Experiments on a 66-language Voice Search task show that we can take a model built on 15 languages and continue training to obtain a 32-language model and similarly to further build a 67-language model. More importantly, models developed in this way achieve better quality compared to those trained from scratch. It maintains similar performance on old languages and achieves competitive results on new ones. This would potentially speed up the development of universal ASR models that recognize speech from any language, any domain and any environment by reusing knowledge learned beforehand.

Index Terms— massive, multilingual, lifelong learning

1. INTRODUCTION

End-to-end (E2E) ASR models that merge the modeling of acoustics, lexicon and language into a single neural network [1–15] largely simplify the building of multilingual models by learning directly from data. Experiments on less than 10 languages have shown promising capabilities 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 [19–22] also show comparable or even better performance than monolingual systems.

Many of these studies mainly focus on developing the best quality model on a fixed multilingual dataset. In practice, scaling up to more languages is usually done incrementally, but there have not been much work investigating how to efficiently adding new languages such that model quality is not degraded. In [23], a configurable multilingual model is developed which can be customized in the inference stage to reduce the serving model size. During training, it still assumed all the languages are known beforehand. [21] investigated the quality affected by similarities between languages used in pretrained models and the target fine-tuning languages. It focuses more on using models pretrained on other languages to boost quality of target languages instead of better ways to scale up to more languages. In this work, we study the impact of adding more languages to the quality of multilingual ASR models. Specifically, we use a dataset consisting of 66 languages in total and create smaller subsets to understand the interaction between the number of languages and the model capacities. Experimental results show that with a fixed model capacity, training with more languages hurts the quality.

To address this quality degradation, we propose to use a lifelong learning approach [24] to transfer knowledge learned from models trained on smaller number of languages. There has been an ongoing paradigm shift with the rise of models that are trained on broad data at scale and are adaptable to a wide range of downstream tasks, which is termed *foundation models* [25]. *Transfer learning* that makes foundation models possible and *scale* that makes them powerful are the two key factors. In the speech community, many efforts have been exploring the use of unsupervised or semi-supervised pretrained models to transfer knowledge learned from a much larger scale of unlabelled datasets to the target tasks [26]. Supervised pretrained ASR models are normally used in natural language processing (NLP) tasks [27]. In NLP, T5 [28] has demonstrated the benefit of transfer learning across large scale supervised NLP tasks. Motivated by that, we investigate the potential of cross-lingual transfer via incrementally adding languages to tackle the challenging language interference problem in a supervised setup. To address the *catastrophic forgetting* [29, 30] phenomenon, we keep the data from existing languages when adding new ones. In summary, the contributions of this paper are as follows:

- We present an experimental study on the effect of adding more languages and its interaction with model capacities in building MMASR models.
- We propose a lifelong learning solution to remedy the language interference problem in building MMASR models by incrementally adding more languages.

[*] Work done in Google.

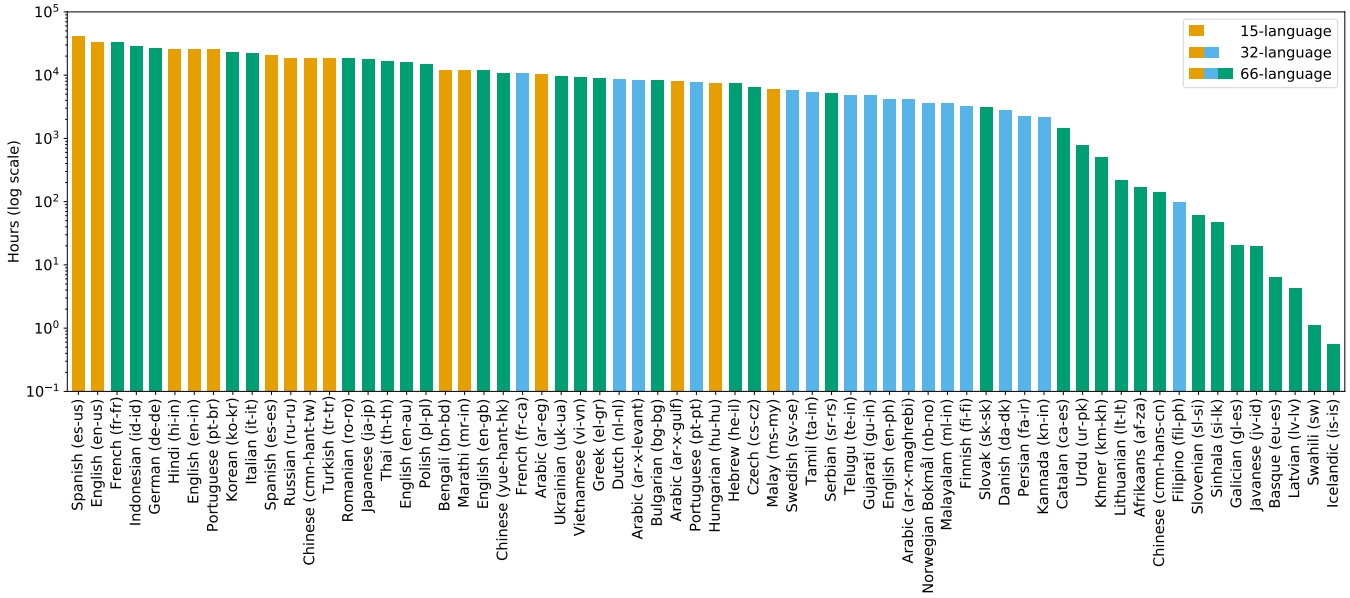


Fig. 1: Per language data distribution. Three subsets are created, namely 15-language set (yellow bars), 32-language set (yellow and blue bars) and 66-language set (yellow, blue and green bars).

2. MMASR SYSTEM

2.1. Data Set

We use a Voice Search dataset consisting of 66 language locales in this study. There are a total of 591.4M utterances which correspond to 670.6K hours of speech data from Voice Search traffic. The data is anonymized and human transcribed. The per language data distribution is depicted in Figure 1. The number of utterances for each language ranges from 800 to 34.6M, roughly corresponding to half an hour to 41.0K hours of speech data. The unbalanced data distribution poses a modeling challenge and the large scale brings in training efficiency challenges. The test set for each language contains from 300 to 23.8K utterances sampled from Voice Search traffic with no overlapping with the training set. Similarly, they are anonymized and human transcribed for evaluation purposes.

To understand the quality implications of adding more languages in the multilingual ASR systems, we create three subsets of languages, namely the 15-language set ($\mathcal{D}_{\text{15-lang}}$), the 32-language set ($\mathcal{D}_{\text{32-lang}}$) and the total 66-language set ($\mathcal{D}_{\text{66-lang}}$). They have the relation of

$$\mathcal{D}_{\text{15-lang}} \subset \mathcal{D}_{\text{32-lang}} \subset \mathcal{D}_{\text{66-lang}}.$$

The detailed list of languages for each subset can be found in Figure 1. The summary statistics of different subsets are listed in Table 1. Briefly speaking, $\mathcal{D}_{\text{15-lang}}$ mainly contains high resource languages and $\mathcal{D}_{\text{32-lang}}$ contains high and medium resource languages [22], while the $\mathcal{D}_{\text{66-lang}}$ brings additional high, medium and also many low resource languages.

Table 1: Total data statistics, including number of utterances in millions (M), total duration in thousand (K) hours and the sizes of the unified grapheme vocabularies.

Data Set	Utterances(M)	Hours(K)	Vocab Size
$\mathcal{D}_{\text{15-lang}}$	231.6	359.2	3328
$\mathcal{D}_{\text{32-lang}}$	312.2	466.4	3712
$\mathcal{D}_{\text{66-lang}}$	591.4	670.6	16858

2.2. Model Architecture

The MMASR systems investigated in this work are attention-based encoder-decoder models. For the encoder, we use full-context Conformer layers [31], 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 others. The remaining Conformer layers make up for the third block. Similar to [32], we use the existing convolution module to provide relative positional information and group normalization to address variations across languages in each Conformer layer. The decoder we adopt is a Transformer decoder with masked self attention and cross attention to the encoder outputs [14, 33]. 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.

Table 2: Configurations for models with different numbers of parameters: 400M (S1), 600M (S2), 800M (S3) and 1B (S4). The exact model size in the table is based on $\mathcal{M}_{\text{66-lang}}$ and is reported in million (M) parameters. "W" for the width of each layer and "L" for the number of layers.

	Encoder			Decoder			Total
	W	L	Size	W	L	Size	Size
S1	768	17	254	768	12	139	394
S2	1024	17	452	768	12	140	592
S3	1024	25	645	768	12	140	785
S4	1024	33	839	768	12	140	979

We use unified multilingual grapheme vocabularies which are generated by pooling all the graphemes from different languages used for training as the output targets. From Table 1, there is a small increase in the output size from $\mathcal{D}_{\text{15-lang}}$ to $\mathcal{D}_{\text{32-lang}}$, but a large jump from $\mathcal{D}_{\text{32-lang}}$ to $\mathcal{D}_{\text{66-lang}}$, mostly due to the addition of many Asian languages such as Korean, Japanese and Thai. Similarly to [16, 22], we feed language information via a one-hot embedding vector into the encoder as an additional input. The dimensions of these language vectors are 16-dimensional(-dim), 48-dim and 80-dim for $\mathcal{D}_{\text{15-lang}}$, $\mathcal{D}_{\text{32-lang}}$ and $\mathcal{D}_{\text{66-lang}}$ respectively.

To understand the quality impact of adding more languages to multilingual models, we build similar systems on the above mentioned subsets. We refer to models trained on subsets $\mathcal{D}_{\text{15-lang}}$, $\mathcal{D}_{\text{32-lang}}$ and $\mathcal{D}_{\text{66-lang}}$ as $\mathcal{M}_{\text{15-lang}}$, $\mathcal{M}_{\text{32-lang}}$ and $\mathcal{M}_{\text{66-lang}}$ respectively. In this study, we simply pool the language specific data from each subset together and sample each training batch according to the natural distribution. In view of the drastic unbalanced data amount across languages, it would be beneficial to adjust the language mixing ratio for training, which will be explored in future work.

2.3. Model Capacity

Based on the findings in [22], larger models have both better quality and faster convergence speed than smaller ones for large-scale multilingual ASR tasks. Taking training efficiency into considerations, a 1B-parameter(-param) model is the largest single model that can currently fit into the on-chip memory of Google's Tensor Processing Unit (TPU) V3 [34] without sharding the models using techniques such as GShard [35]. We hence in this study cap our model capacity at 1B parameters. Previously [22], we found that allocating a fixed capacity to encoders tends to work better than decoders. We hence adopt a 12-layer Transformer decoder with model dim of 768 and hidden dim of 3072, and fix it across different models. This decoder corresponds to 139M parameters. For encoders, we create 4 different configurations which roughly correspond to a total number of model parameters of 400M (S1), 600M (S2), 800M (S3) and 1B (S4). The detailed encoder architectures are listed in Table 2.

2.4. Model Training

We use 80-dim 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 240-dim input representation with 30ms frame rate. A one-hot language vector is appended to the log Mel features before passing into the encoder. 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 using the Lingvo [37] toolkit on Google's Tensor Processing Units (TPU) V3 [34] with a global batch size of 4,096 utterances. Models are trained with 512 TPU cores and optimized using synchronized stochastic gradient descent. Adafactor [38] with parameters β_1=0.9 and β_2=0.99 is used. A transformer learning rate schedule [33] with peak learning rate 3e-4 and 10K warm-up steps is used. Exponential moving average is used to stabilize the model weight updates.

3. RESULTS

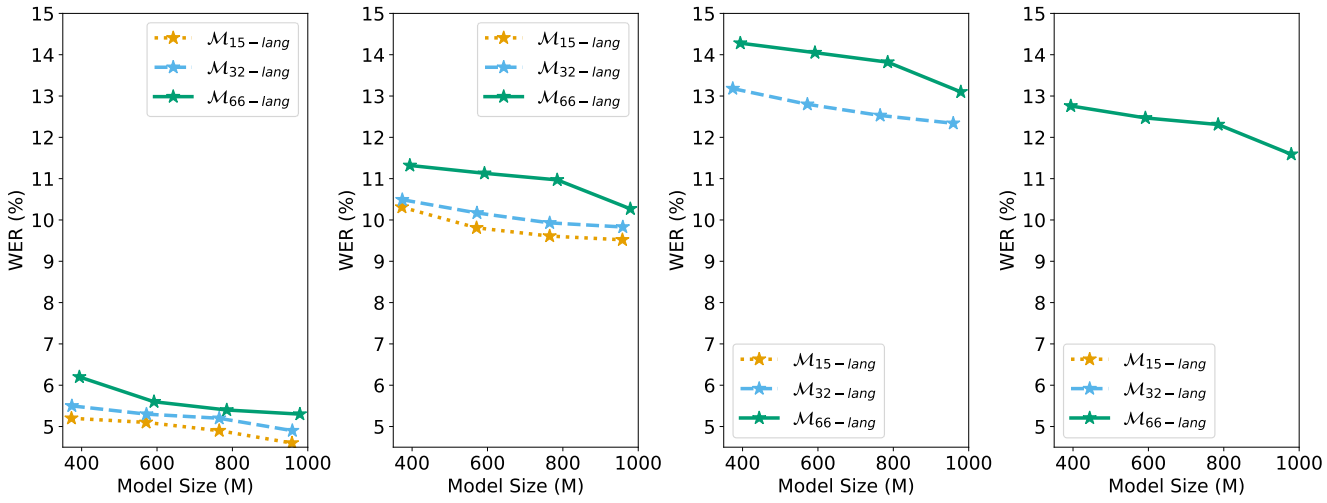
In this section we present our study of building MMASR models on different subsets. We use average word error rate (WER) with equal weights across languages for comparisons.

3.1. Effects of adding more languages

To understand the impact of adding more languages to multilingual models, we trained model $\mathcal{M}_x$ on $\mathcal{D}_x$ for each of the subsets described in Section 2.1. To further understand how the model size affects that relationship, for each $\mathcal{M}_x$, we trained 4 different sizes as described in Section 2.3. The results are depicted in Figure 2. Firstly, we look at the performance difference on a single language test set - English (en-us) (Figure 2a). Larger model always yields better performance. When comparing models trained on different subsets, quality degradation is observed after adding more languages.

Next, looking at the average WER on the $\mathcal{D}_{\text{15-lang}}$ subset (Figure 2b), similar trends are observed: larger model reduces WER and more languages hurts the quality. Specifically, the best average WER on the $\mathcal{D}_{\text{15-lang}}$ test sets is obtained by the model $\mathcal{M}_{\text{15-lang}}$. Once more languages are added (such as $\mathcal{M}_{\text{32-lang}}$), the quality degrades given the same model capacity. However, if we increase the model capacity while adding more languages, we can maintain similar performance. For example, to match the quality of the 400M-param $\mathcal{M}_{\text{15-lang}}$ on the $\mathcal{D}_{\text{15-lang}}$ test sets, we need to use a 600M-param $\mathcal{M}_{\text{32-lang}}$ or a 1B-param $\mathcal{M}_{\text{66-lang}}$. Similarly on the $\mathcal{D}_{\text{32-lang}}$ test sets (Figure 2c), the $\mathcal{M}_{\text{66-lang}}$ needs to use 1B parameters to ensure the same quality as the 400M-param $\mathcal{M}_{\text{32-lang}}$.

Lastly, when comparing the $\mathcal{M}_{\text{66-lang}}$ across Figure 2b, Figure 2c and Figure 2d, the average WER increases from $\mathcal{D}_{\text{15-lang}}$ test sets to $\mathcal{D}_{\text{32-lang}}$ test sets; while it decreases from $\mathcal{D}_{\text{32-lang}}$ test sets to $\mathcal{D}_{\text{66-lang}}$ test sets. This could be possibly



(a) On English (en-us) test set. (b) On $\mathcal{D}_{15\text{-lang}}$ Test sets. (c) On $\mathcal{D}_{32\text{-lang}}$ Test sets. (d) On $\mathcal{D}_{66\text{-lang}}$ Test sets.

Fig. 2: WER (%) comparisons of models trained on different number of languages ($\mathcal{M}_{15\text{-lang}}$, $\mathcal{M}_{32\text{-lang}}$ and $\mathcal{M}_{66\text{-lang}}$), with different sizes (400M, 600M, 800M and 1B) on four different test sets: (a) a single language English (en-us), (b) $\mathcal{D}_{15\text{-lang}}$ test sets, (c) $\mathcal{D}_{32\text{-lang}}$ test sets and (d) $\mathcal{D}_{66\text{-lang}}$ test sets. We use average WER when there is more than one language. The comparison is done at 200K training steps for all the models.

related to the specific languages added each time. One possible factor is that when we move from $\mathcal{D}_{32\text{-lang}}$ to $\mathcal{D}_{66\text{-lang}}$, there are many low resource languages added into training, which normally benefit more from multilingual training.

3.2. Lifelong Learning

Training multilingual models each time from scratch can become a painful process especially when the target set of languages changes. In previous work [22], we found positive knowledge transfer in supervised trained models when using them to seed training on more languages. In this section, we further justify this finding by training the 1B-param model firstly on a small set of languages and then continuing training with more languages added. Due to the change of the input language ID vector and the output vocab size, we drop existing models' input projection and output layer, and initialize them with random values. As a baseline, we train a 1B-param 66-language model from scratch ($\mathcal{M}^{\text{rand}}_{66\text{-lang}}$ in Table 3). Comparing with initializing from an existing 1B-param $\mathcal{M}_{15\text{-lang}}$ to seed the training, namely $\mathcal{M}^{\text{init-15}}_{66\text{-lang}}$, at the same 200K training steps, we get much better results by reusing the old model. We further split the training process into three stages, namely first train from scratch a 1B-param $\mathcal{M}^{\text{rand}}_{15\text{-lang}}$; then we take the converged model and continue training on $\mathcal{D}_{32\text{-lang}}$ leading to $\mathcal{M}^{\text{init-15}}_{32\text{-lang}}$; after that we take the converged $\mathcal{M}^{\text{init-15}}_{32\text{-lang}}$ to seed the training on 66 languages, yielding $\mathcal{M}^{\text{init-32}}_{66\text{-lang}}$. The final results are presented in Table 3. With lifelong learning, we can maintain the better performance of models trained with smaller number of languages even when scaling up to more languages without increasing model capacity. This addresses the quality degradation observed in Figure 2, namely, given a fixed capacity, adding more languages increases WERs. Also when comparing against monolingual models, all the multilingual models are better than monolingual models in terms of quality. We built up to only 32 monolingual models due to the large amount of tuning needed.

Table 3: WER (%) performance of 1B-param MMASR models trained in a lifelong learning setup. Each monolingual model has 140M parameters with WER at convergence.

Exp.	**en-us**	$\mathcal{D}_{\textbf{15-lang}}$	$\mathcal{D}_{\textbf{32-lang}}$	$\mathcal{D}_{\textbf{66-lang}}$
Monolingual	4.6	9.3	11.9	-
$\mathcal{M}^{\text{rand}}_{66\text{-lang}}$	5.3	10.3	13.1	11.6
$\mathcal{M}^{\text{init-15}}_{66\text{-lang}}$	4.7	9.6	12.4	11.0
$\mathcal{M}^{\text{rand}}_{15\text{-lang}}$	4.3	9.1	-	-
$\mathcal{M}^{\text{init-15}}_{32\text{-lang}}$	4.2	9.2	11.6	-
$\mathcal{M}^{\text{init-32}}_{66\text{-lang}}$	4.4	9.1	11.8	10.4

4. CONCLUSIONS

In this paper, we studied the impact of adding more languages to multilingual model performance. On a 66-language dataset we see clear quality degradation when adding more languages without changing the model capacity. Increasing capacity improves model quality, but the gap caused by more languages remains. A lifelong learning approach is developed, which reuses models trained on smaller sets of languages and incrementally adds more languages. With this we can maintain similar performance on existing languages with a fixed capacity while scaling up to more languages.

5. REFERENCES

[1] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," in *Proc. NIPS*, 2015.

[2] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Proc. Interspeech*, 2015.

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

[4] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013.

[5] E. Battenberg, J. Chen, R. Child, A. Coates, Y. Gaur, Y. Li, H. Liu, S. Satheesh, D. Seetapun, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*, 2017.

[6] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proc. ASRU*, 2019.

[7] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A new training pipeline for an improved neural transducer," in *Proc. Interspeech*, 2020.

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

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

[10] C.-C. Chiu, T. N. Sainath, Y. Wu, et al., "State-of-the-art Speech Recognition With Sequence-to-Sequence Models," in *Proc. ICASSP*, 2018.

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

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

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

[14] Q. Zhang, H. Lu, H. Sak, et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. ICASSP*, 2020.

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

[16] B. Li, T. N. Sainath, K. C. Sim, et al., "Multi-Dialect Speech Recognition With A Single Sequence-To-Sequence Model," in *Proc. ICASSP*, 2018.

[17] A. Kannan, A. Datta, T. N. Sainath, et al., "Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model," in *Proc. Interspeech*, 2019.

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

[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] B. Li, R. Pang, T. N. Sainath, et al., "Scaling End-to-End Models for Large-scale Multilingual ASR," in *Proc. ASRU*, 2021.

[23] L. Zhou, J. Li, E. Sun, and S. Liu, "A configurable multilingual model is all you need to recognize all languages," *arXiv preprint arXiv:2107.05876*, 2021.

[24] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[25] R. Bommasani, D. A. Hudson, E. Adeli, et al., "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[26] Yu Z., Daniel S. P., Wei H., et al., "BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition," *arXiv preprint arXiv:2109.13226*, 2021.

[27] C.-I. Lai, Y.-S. Chuang, H.-Y. Lee, S.-W. Li, and J. Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," in *Proc. ICASSP*, 2021.

[28] C. Raffel, N. Shazeer, A. Roberts, et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[29] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[30] Y. Bengio, M. Mirza, I. Goodfellow, A. Courville, and X. Da, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," 2013.

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

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

[33] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, 2017.

[34] "Cloud Tensor Processing Units (TPUs," `https://cloud.google.com/tpu/docs`, Accessed: 2021-09-30.

[35] D. Lepikhin, H.-J. Lee, Y. Xu, et al., "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[36] D. S. Park, W. Chan, Y. Zhang, et al., "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," *Interspeech 2019*, Sep 2019.

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

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