Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters

Vineel Pratap[1], Anuroop Sriram[1], Paden Tomasello[1], Awni Hannun[2], Vitaliy Liptchinsky[1],
Gabriel Synnaeve[2§], Ronan Collobert[1§]

[1]Facebook AI Research, Menlo Park
[2]Facebook AI Research, NYC

{vineelkpratap,anuroops,padentomasello,awni,vitaliy888,gab,locronan}@fb.com

Abstract

We study training a single acoustic model for multiple languages with the aim of improving automatic speech recognition (ASR) performance on low-resource languages, and overall simplifying deployment of ASR systems that support diverse languages. We perform an extensive benchmark on 51 languages, with varying amount of training data by language (from 100 hours to 1100 hours). We compare three variants of multilingual training from a single joint model without knowing the input language, to using this information, to multiple heads (one per language "cluster"). We show that multilingual training of ASR models on several languages can improve recognition performance, in particular, on low resource languages. We see 20.9%, 23% and 28.8% average WER relative reduction compared to monolingual baselines on joint model, joint model with language input and multi head model respectively. To our knowledge, this is the first work studying multilingual ASR at massive scale, with more than 50 languages and more than 16,000 hours of audio across them.

Index Terms: speech recognition, multilingual

1. Introduction

The use of multilingual ASR systems [1, 2, 3, 4, 5, 6, 7, 8, 9] that simultaneously transcribe multiple languages has recently become popular to increase language coverage, but covering all of the world's ∼7000 languages is still far ahead. The ability to train a single model on many, say more than 50 languages, presents a few advantages. First, in a production setting, training, deploying and maintaining one model per language, especially on long tail of low resource languages, can quickly become cumbersome as the number of languages increases. Having a single model for all languages can simplify the production pipeline significantly. Second, as previously shown in the literature, training multilingual ASR models on a small set of similar languages can improve recognition performance. However, it is not clear if these multilingual approaches can scale to a large number of diverse languages, from different language families.

In this work, we (i) train at scale on 51 languages from several language families, and (ii) show that a joint model with shared vocabulary approach can surpass strong monolingual baselines. Furthermore, (iii) we propose a refined multi-head approach, where each head addresses a set of similar languages and improves on the monolithic joint model approach, leading to competitive results (compared to monolingual baselines) also on higher-resource languages. Finally, (iv) we demonstrate that our multilingual model learns representations general enough that it improves monolingual baseline WER on new languages not seen during the initial training phase.

_____

§Equal Advising

2. Related Work

A single model capable of recognizing multiple languages has been a long-term goal in the field of automatic speech recognition. Models capable of learning from multiple languages have been studied in both the HMM-GMM [1, 3] and DNN-HMM hybrid systems [4]. In general, multi- and cross-lingual speech processing has been an active area of research for decades [5].

More recently, as end-to-end models have matured in the monolingual setting [10, 11], attention has turned to leverage multiple languages to further improve their performance. End-to-end models typically require more data to match and surpass the performance of hybrid systems and thus leveraging data from multiple languages is more relevant now than ever.

Multilingual sequence-to-sequence (seq2seq) models have been shown to improve performance in the cross-lingual setting [6], where the model is first pre-trained on a language group and then fine-tuned to a specific target language. Prior work has leveraged as many as 100 languages simultaneously to learn language agnostic features [12]. The work, however, studied a limited dataset which consists of Bible readings in different languages, with only a single reading of Bible for most languages.

In [7] and [8], authors train a seq2seq and RNN-T [13] models respectively on 9 Indian languages. Both approaches perform analysis using shared encoder, decoder with language identification as an additional input and train with unified grapheme set from all languages. While the former approach trains a single model, the latter one uses additional adaptive per-language layers. In [9], an acoustic model outputs Unicode bytes directly rather than letters or sentence pieces. They show gains in the multilingual setting and espouse the efficiency benefits of predicting bytes in avoiding large Softmax layers.

All of the discussed related work either study limited set of languages, typically less than 10, or limited data, such as readings of Bible. To the best of our knowledge, this work is the first one to study multilingual systems at a massive scale, covering 51 languages and more than 16,000 hours of audio.

3. Multilingual models

3.1. Seq2Seq Model

A sequence-to-sequence (Seq2Seq) model comprises of two neural networks: an encoder and a decoder. The encoder maps the input audio sequence $\mathbf{x} = (x_1, ..., x_T)$ to an intermediate representation $\mathbf{h} = (h_1, ..., h_K)$. The decoder maps $\mathbf{h}$ to the output sequence $\mathbf{y} = (y_1, ..., y_L)$ in an autoregressive manner. Specifically, we use a stacked unidirectional RNN that computes the probability of the sequence $\mathbf{y}$ using:

$$P(\mathbf{y}|\mathbf{h}) = \prod_t P(y_t|\mathbf{h}, \mathbf{y}_{<t}) \qquad (1)$$
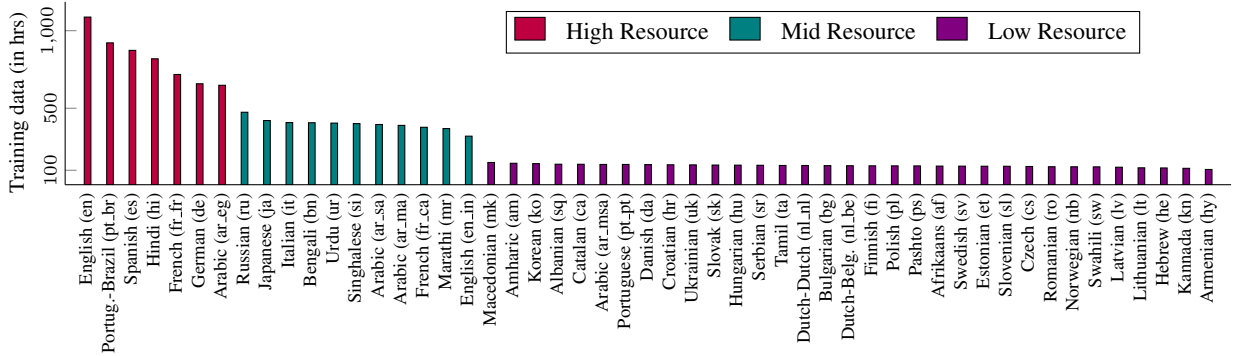
Figure 1: *Training data distribution across different languages*

Thus, the decoder learns a language model conditioned on the hidden representations **h**. The encoder and decoder networks are jointly optimized to minimize the cross-entropy loss between the output of the network and the ground truth transcriptions. In our models, the encoders are based on the time-depth separable convolution (TDS) architecture [14].

For multilingual training, we consider N languages $(\mathcal{L}_1, ..., \mathcal{L}_N)$ with each language $\mathcal{L}_i$ consisting of an independent training set $\{\mathcal{X}_i, \mathcal{Y}_i\}$ comprising of n_i samples. Each language $\mathcal{L}_i$ has a set of graphemes $\mathcal{G}_i$ that may overlap with the graphemes from other languages. We train all of our multilingual models on the combined training set $\cup_{i=1}^{N}(\{\mathcal{X}_i, \mathcal{Y}_i\})$.

3.2. Shared sub-word tokens

Working with a large set of languages, each with their own distinct character set and tokenization rules, makes training and maintaining models cumbersome. Adding or removing languages would require modifications to the model architecture and training routine for example. To simplify this process, we create a shared token set across all languages using a Sentence Piece Model (SPM) [15] . Similar to [16], the shared sentence pieces are built by sampling the sentences using a multinomial distribution $\{s\}_{i=1..N}$,

$$s_i = \frac{p_i^\alpha}{\sum_{j=1}^{N} p_j^\alpha} \quad \text{with} \quad p_i = \frac{n_i}{\sum_{k=1}^{N} n_k}, \qquad (2)$$

where the parameter α controls the sampling of languages with different frequencies.

3.3. Joint model

Our joint model approach is a single model which is trained while sharing the parameters from the encoder, decoder and token set, across all languages. We optionally (see Section 5) feed language information to the model in the form of an embedding, which is also trained jointly with the ASR model.

3.3.1. Curriculum training of joint model

We faced convergence issues with joint model, when training on data from all languages. For these cases, we introduced a curriculum training [17] based approach, which incrementally adds each language after the model has been trained for a fixed number of iterations or the Character Error Rate (CER) goes below 50% for the previously added language. We found that training converges easily for up to 51 languages using this method.

Group name	Languages
Latin	af, ca, da, de, en, en_in, es, et, fi, fr_ca, fr_fr, hu, it, lt, nl_be, nl_nl, pt_br, pt_pt, ro, sq, sv, sw
Balto-Slavic	cs, hr, lv, nb, pl, sk, sl
Indic	bn, hi, kn, mr, si, ta
Perso-Arabic	am, ar_eg, ar_ma, ar_msa, ar_sa, he, ps, ur
Cyrillic	bg, mk, ru, sr, uk
Misc	hy, ja, ko

Table 1: *Language groups used for the multi-headed models*

3.4. Multi-headed model

Joint training of multiple tasks can only be beneficial when the individual tasks share common representations. Since the decoder of a seq2seq model learns a conditional language model, sharing decoder parameters between languages that do not have any graphemes in common is unlikely to improve the recognition performance of any of the languages. Therefore, we divide the languages into M distinct groups and use a different decoder for each language group. Thus, our multi-headed models employ a single encoder with shared parameters, and a set of M decoders, one per language group. In the forward pass, the appropriate decoder is selected based on the language.

Ideally, we would like to group the languages by their written scripts. However, this leads to a skewed distribution of group sizes, with a few language groups containing many languages and others containing only a single language. In such a setting, it becomes critical to tune the decoder hyperparameters (like the number of RNN layers) for each language group, adjusting the head capacity according to the amount of training data available for that group. To avoid this additional complication, we manually combined some of the smaller language groups together until we end up with six language groups. The language groups we used in our experiments are shown in Table 1. We do not use curriculum training (Section 3.3.1 for multi-headed models because they are able to converge even when trained with all 51 languages together.

4. Experimental details

4.1. Data

The training set used for our experiments consists of videos publicly shared by users that are anonymized and spans a total of 51 languages. Figure 1 shows the amount of training data present in all the languages. We categorize the languages into three categories – high resource languages consisting of >600

hours of train data, mid resource language with 300-500 hours of training data and low resource languages with 100-150 hrs of training data. We use about 20 hours of test set for each language. All hyper parameter tuning is done on a held-out development set of about 13 hours of high and mid resource languages, and about 7 hours for low resource languages.

Since our dataset is transcribed with predefined guidelines, we were able to avoid many nuances which can arise when mining the text from online sources. For each language, we normalize the text by performing NFKC normalization and removing all punctuations. We then prepare a list of valid unicode characters based on the language's orthography and filter words which contain characters outside this range. We use this data for generating the token set and lexicon as well as model training.

4.2. Training setup

All our experiments are run using wav2letter++ [18] framework. We use 80-dimensional log-mel features as input, with STFTs computed on 30ms hamming windows strided by 10ms. All our acoustic models are based on the system proposed in [14]. We use SpecAugment [10] for all our experiments with LibriSpeech Double setting. We also use Blockwise Momentum Update Filtering (BMUF) [19] for all the experiments to help with scaling the training workflows.

4.3. Monolingual baseline models

All baseline models use an encoder with three 10-channel, four 14- channel and eight 18-channel Time Depth Separable Convolution (TDS) [14] blocks. We use three 1D convolutions to subsample over time, one as the first layer and one in between each group of TDS blocks. Each 1D convolution module has a stride of 2 which accounts for a total sub-sampling factor of 8. We use kernel size of 21 for all the convolution layers. These layers are followed by a final Linear layers with 1024-dimensional encoder output. The decoder which is also based on [14] consists of two-layer GRU with 512 hidden units with two rounds of inner-product key-value attention. Overall, the combined encoder and decoder model has about 150 Million parameters.

We have tuned dropout and hyper parameters in BMUF extensively for all the models. For the high and mid resource languages, we use 5K and 2K shared sentencepiece tokens. For low resource languages, we use graphemes as the modelling units as it gave better performance over sub-word units. For all languages, the test WER is taken for the epoch which produces the best validation WER.

4.4. Training data sampling for multilingual models

Because of the imbalance of data across languages, it can be difficult for models to perform well on low resource languages. Similar to [8], we sample data from a language $\mathcal{L}_i$ during training from a multinomial distribution $s_{i=1..N}$ as given below

$$s_i = \frac{n^{max} + \beta * (n_i - n^{max})}{\sum_{j=1}^{N} n^{max} + \beta * (n_j - n^{max})} \quad (3)$$

where n^{max} is the maximum number of training samples across any language and β is a tunable parameter that allows us to adjust sampling of languages from their natural frequency, when $\beta = 1$, to a uniform distribution across languages when $\beta = 0$.

5. Results and analysis

In this section, we present our experimental results on three multilingual models - joint model, joint model with language

input and multi-head model described in Section 3.

5.1. Study of tunable parameters α, β

As mentioned in section 3.2 and 4.4, we use tunable parameters α, β for controlling the sampling of languages during token generation and training examples during multilingual model training respectively. We compare the WER performance of a 500 million parameter joint model with varying sampling fractions and the results are presented in Figure 2.
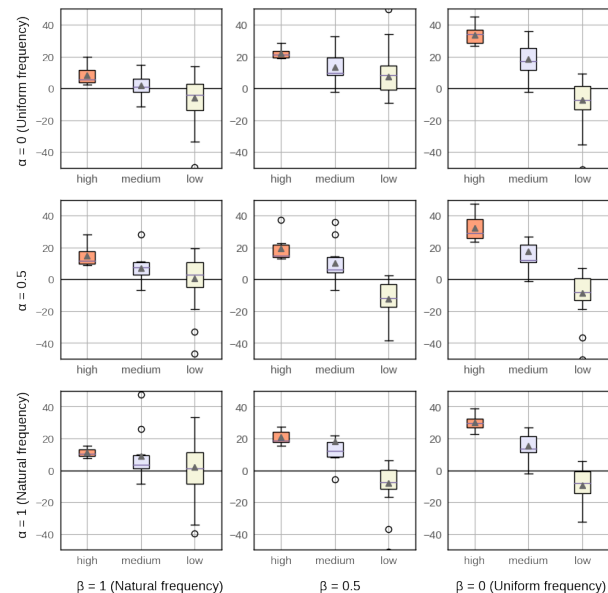

Figure 2: *Box plots of relative WER change from monolingual baseline (lower is better) on a 500 million joint model using 10K shared sentence pieces with varying α and β. As an example, $\alpha = 0$ and $\beta = 0$ represents sampling from all languages uniformly for both vocabulary creation and multilingual model training. Languages are sectioned by their resource category.*

In general, we see that going from natural frequency ($\alpha = 1$, $\beta = 1$) to uniform frequency ($\alpha = 0$, $\beta = 0$) seems to improve performance of low resource languages while degrading performance on high resource languages. Interestingly, it appears the using $\alpha = 0.5$ and $\beta = 0.5$ performs best on low resource languages and has less performance degradation on high, mid resource languages compared to sampling at uniform frequency ($\alpha = 0$ and $\beta = 0$). For low resource language, one might assume that sampling a language more frequently will always result in better performance. We believe that sampling at the natural frequency has too much data imbalance to learn an effective shared representation, while sampling at the uniform distribution overfits to the low resource languages. We use $\alpha = 0.5$ and $\beta = 0.5$ for all our multilingual experiments.

5.2. Joint model

We use a shared token set of 10K sentence pieces for all joint model experiments. We have also tried shared graphemes and shared sentence pieces of size 25K, 50K and empirically found that 10K sentence pieces give the best performance. For the joint model with input language embedding, we use a 10-dimensional language embedding and concatenate it with the log-mel input features and feed it to the encoder.

(a) Joint model

(b) Joint model with input language embedding

- ■ 150 Mil ● 500 Mil ▲ 1 Bil

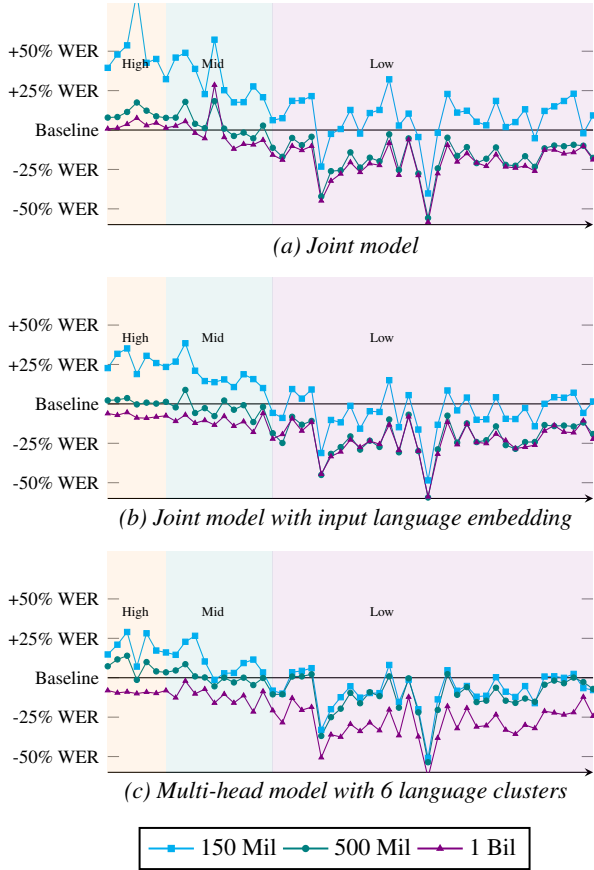*(c) Multi-head model with 6 language clusters*

Figure 3: *Relative WER change from monolingual baseline (lower is better) for different multilingual models as we increase model size. The amount of training data is gradually reducing as we move along x-axis for each plot. For 'ja' and 'ko', we use Character Error Rate (CER) instead of Word Error Rate (WER)*

As mentioned in Section 3.3.1, we use curriculum training for training the models. SpecAugment[10] is applied once all the languages added in the curriculum training. Figure 3(a) shows the results of joint model and Figure 3(b) shows the results of joint model with input language embedding for different model sizes. We can see that increasing the model size helps with improving WER in both settings.

For 1 billion parameter joint model, we see an average WER degradation of 3.2% on high resource and an average WER improvement of 2.5% and 20.9% on mid and low resource languages. Further, we observe that using language embedding at the input layer performs better that without using it for a given model size. For 1 billion parameter model with language input, we observe an improvement in WER on all the languages. The average WER improvements on high, mid and low resource languages are 7.5%, 12.1% and 23% respectively.

5.3. Multi-head model

We select 10K subword units as the token set for each language group as described in section 3.2. Figure 3(c) shows the relative change in WER obtained with multi-headed models of different sizes compared to the baseline for each language. The largest multi-headed model with 1 billion parameters can significantly improve performance on all languages. This model improves WER by 9.1% on average for high-resource languages, by 12.4% for mid-resource languages, and by 28.8% for low-
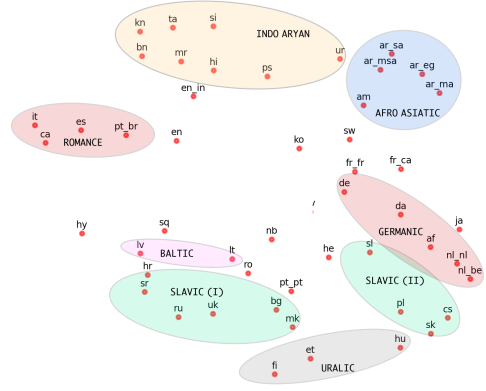

Figure 4: *t-SNE clustering of trained language embeddings. Colored clusters are based on language families.*

resource languages. The largest multi-headed model also outperforms the joint models even when the joint models are fed a language embedding. In addition, the multi-headed models are simpler to train as they do not need curriculum training.

5.4. Multilingual transfer learning on unseen languages

Training multilingual models on a large, diverse set of languages enables the acoustic models to learn language-agnostic representations general enough to perform well on completely new languages. To demonstrate this, we fine-tune the joint model with 1 billion parameters on 3 unseen low-resource languages. Since, the graphemes in new languages, which are being fine-tuned, are not present in the decoder of trained joint model, we re-initialize the decoder for the grapheme set of new language. We allow both encoder and decoder to be trained during fine-tuning.

From table 2, we can see that fine-tuning on joint model improves the WER over monolingual baselines. The fine-tuning approach can thus help with adapting a new language easily while also improving the WER from monolingual baseline.

Language	Monolingual WER	Fine-tune WER
Chinese (zh_tw)	50.82	39.29 (-22.7%)
Persian (fa)	33.59	31.29 (-6.8%)
Telugu (te)	50.05	47.63 (-4.8%)

Table 2: *Fine-tuning results on joint model. For 'zh_tw', we report Character Error Rate (CER) instead of WER*

5.5. Language embedding analysis

We use t-SNE [20] method to visualize the learned language embeddings of the joint model with language input. From Figure 4, we can notice most of the learned embeddings form noticeable clusters for language families. Similar to [21], these learned clusters can be used for training multi-head experiments instead of choosing the clusters manually.

6. Conclusion

We demonstrated that it is possible to train a massive single ASR architecture for 51 various languages, which we found in practice considerably less time-consuming to tune than 51 different monolingual baselines. Compared to the monolingual models, our multilingual approach improves WER performance on all languages, which further consolidates its advantage.

7. References

[1] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, D. Povey *et al.*, "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 4334–4337.

[2] 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," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2018.8461886

[3] H. Lin, L. Deng, D. Yu, Y.-f. Gong, A. Acero, and C.-H. Lee, "A study on multilingual acoustic modeling for large vocabulary asr," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 4333–4336.

[4] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8619–8623.

[5] H. Bourlard, J. Dines, M. Magimai-Doss, P. N. Garner, D. Imseng, P. Motlicek, H. Liang, L. Saheer, and F. Valente, "Current trends in multilingual speech processing," *Sadhana*, vol. 36, no. 5, pp. 885–915, 2011.

[6] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiat, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 521–527.

[7] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4904–4908.

[8] 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," *Interspeech 2019*, Sep 2019.

[9] 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.

[10] 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

[11] G. Synnaeve *et al.*, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *preprint arXiv:1911.08460*, 2019.

[12] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, "Massively multilingual adversarial speech recognition." Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 96–108.

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

[14] A. Hannun, A. Lee, Q. Xu, and R. Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," in *INTERSPEECH*, 2019.

[15] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," 2018.

[16] A. Conneau and G. Lample, "Cross-lingual language model pretraining," in *Advances in Neural Information Processing Systems*, 2019, pp. 7057–7067.

[17] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[18] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6460–6464.

[19] K. Chen and Q. Huo, "Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5880–5884.

[20] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[21] X. Tan, J. Chen, D. He, Y. Xia, T. Qin, and T.-Y. Liu, "Multilingual neural machine translation with language clustering," 2019.