ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747543

EXPLORING EFFECTIVE DATA UTILIZATION FOR LOW-RESOURCE SPEECH RECOGNITION

Zhikai Zhou, Wei Wang, Wangyou Zhang, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{zhikai.zhou,wangwei.sjtu,wyz-97,yanminqian}@sjtu.edu.cn

ABSTRACT

Automatic speech recognition (ASR) has suffered great performance degradation when facing low-resource languages with limited training data. In this work, we propose a series of training strategies to exploring more effective data utilization for low-resource speech recognition. In low-resource scenarios, multilingual pretraining is of great help for the above purpose. We exploit relationships among different languages for better pretraining. Then, the knowledge extracted from the language classifier is utilized for data weighing on training samples, making the model more biased towards the target low-resource language. Moreover, dynamic curriculum learning as a warm-up strategy and length perturbation as data augmentation are also designed. All these three methods form a newly improved training strategy for low-resource speech recognition. Meanwhile, we evaluate the proposed strategies using rich-resource languages for pretraining (PT) and finetuning (FT) the model on the target language with limited data. The experimental results show that on the CommonVoice dataset, compared with the commonly used multilingual PT+FT method, the proposed strategies achieve a relative 15-25% reduction in word error rate on different target languages, which shows the significant effects of the proposed data utilization strategy.

***Index Terms*—** low-resource speech recognition, curriculum learning, data augmentation, data utilization.

1. INTRODUCTION

Automatic speech recognition (ASR) systems need numerous hours of transcribed speech to achieve a fine performance. However, there are more than 6,000 languages in the world. Most of them have been suffering from the insufficiency of the annotated data. For many languages, only a little annotated data are available.

In low-resource scenarios, cross-lingual transfer learning is broadly adopted in many works for hybrid systems [1, 2, 3] and end-to-end systems [4]. Meanwhile, end-to-end ASR models avoid the pronunciation modeling required for hybrid systems.

Then, modeling units and data augmentation for low-resource scenarios have been extensively studied. Articulatory attributes are general for all human languages so that they are adopted as modeling units in works [5, 6]. Transliterations of different languages have been treated as training samples for multilingual data augmentation [7]. LRSpeech [8] has also adopted text-to-speech (TTS) based data augmentation and dual transformation.

On the other hand, self-training and self-learning methods have been proposed to exploit unlabeled data. Noisy student training [9, 10] has predicted hypotheses on unlabelled data. Then they trained the model on augmented data with pseudo labels. Masked acoustic models [11, 12] utilized self-learning for predicting the masked part of speech samples. Recently, in wav2vec 2.0 [13], contrastive learning and masked acoustic models were both utilized for self-learning.

The existing methods above mainly focus on different training paradigms and the utilization of unlabeled data. However, training strategies or ways such as weighing, scheduling, and augmentation are also important perspectives. For multilingual ASR pretraining, previous works only combined the data simply from different languages without considering correlations among languages. In this work, we make three main contributions as follows. Firstly, the data weighing method based on utterance level language similarity is proposed and evaluated. Such similarities are exploited for better adaptation of low-resource ASR. Secondly, novel warm-up strategy based on dynamic curriculum learning method is designed to exploit the data scheduling scheme for making the model be better optimized. The order and usage of training samples are revised, and sample difficulties and model competence are taken into consideration. Thirdly, a novel data augmentation approach named "length perturbation" is developed for end-to-end low-resource ASR. It generates new samples based on utterance fragments and can also be combined with the existing data augmentation methods.

Experimental results show the integration of our methods outperforms the multilingual pretraining + finetuning baseline with a relative 15-25% word error rate reduction.

2. MULTILINGUAL PRETRAINING AND FINETUNING FOR LOW-RESOURCE ASR

2.1. Transformer-based E2E ASR

Transformer is a sequence-to-sequence (S2S) network constructed with an encoder and a decoder network. Each transformer module consists of a multi-head self-attention and several fully connected feed-forward layers [14]. The transformer model is trained under the joint connectionist temporal classification (CTC)/attention (ATT) framework to improve robustness and achieve fast convergence [15]. Denote by $\mathcal{L}_{ctc}$ and $\mathcal{L}_{s2s}$ the CTC and S2S objectives, the loss function of the joint CTC-attention is defined as:

$$\mathcal{L}_{jca} = \lambda \mathcal{L}_{ctc} + (1 - \lambda)\mathcal{L}_{s2s} \tag{1}$$

The tunable coefficient $\lambda \in [0, 1]$ controls the contribution of losses. Joint CTC/ATT decoding is adopted to predict the output sequence.

[†]Yanmin Qian is the corresponding author.

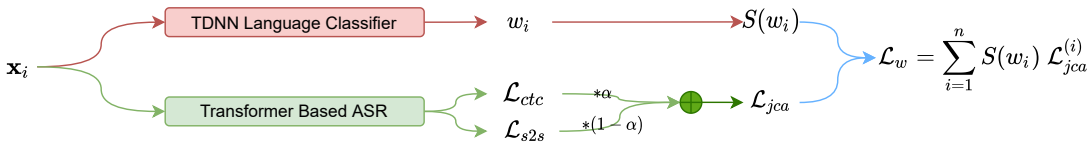

Fig. 1: The proposed data weighing pipeline

2.2. Multilingual Pretraining and Finetuning

Due to the similarities of pronunciation and grammars among human languages, the multilingual pretrained models can learn common speech and language knowledge well [4]. Since many languages already have had a large amount of data, the E2E ASR model has been firstly pretrained on several rich-resource languages. Hereafter, we finetune the ASR model on a low-resource language. The modeling units are from both rich- and low-resource languages. The common knowledge among different languages can be transferred to the low-resource ASR model by the pretrained parameters.

3. OPTIMIZED DATA UTILIZATION FOR LOW-RESOURCE SPEECH RECOGNITION

Most works are merely to train the model towards the whole training set epoch by epoch. In low-resource scenarios, how to better use data is a key problem worth exploration. To improve the performance of low-resource speech recognition, data weighing based on language similarities, warm-up strategy based on dynamic curriculum learning, and data augmentation based on length perturbation are proposed in this work.

3.1. Data Weighing Based on Language Similarity

Multilingual data have been simply combined in many works [16, 17]. However, the correlation and similarity among languages are ignored in those works. For example, the similarity of spelling and pronunciation between French and Catalan is much greater than that between French and Chinese. In previous works, language-level similarity is used from data selection to extract the tandem feature[18, 19, 20] in hybrid systems and phoneme recognition systems. The above mentioned idea is further extended to explore the utterance-level language similarity and it benefits to low-resource language modeling of end-to-end ASR architecture in this work.

3.1.1. Data Weighing

In order to utilize the similarities among languages for ASR training, the data weighing method is proposed. The purpose for using language similarities is to find more data similar to the target language in the multilingual dataset for better adaptation. To obtain such similarities, a method is to exploit the knowledge from a language classifier. The posterior of the target language from the classifier can be considered with language similarities from the model's perspective, which is used as weights of utterances in multilingual pretraining.

Figure 1 shows the pipeline of the proposed method. The weights from the language classifier are firstly extracted on each utterance shown on the top of the figure. Then the loss of each utterance is multiplied with weights to make the model pay more attention to utterances with greater similarities.

3.1.2. Weights Calculation

The posterior of the target language can be considered as the weight of towards the target language, which can be computed as follows.

$$w_i = P(y = l|x_i) \tag{2}$$

where x_i is the input feature of the sample i, l refers to the target language, and $P(y = l|x_i)$ is the posterior from the softmax layer.

The posteriors are sometimes close to one-hot vectors such that the weights of the non-target languages are too small. Inspired from the speaker verification task [21], we also try to extract the embeddings from hidden layers. Embeddings from languages are averaged as the language center. The weights w_i are cosine similarities between language embedding center and utterance embeddings in Equation 4.

$$cos\_sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||\,||\mathbf{b}||} \tag{3}$$

$$w_i = \frac{1 + cos\_sim(s_i, \frac{\sum_{k=1}^{M} s_k}{M})}{2} \tag{4}$$

where $cos\_sim(\mathbf{a}, \mathbf{b})$ is the cosine similarity, s_i is the embedding of sample i. $s_k \in L$ where L is the set of target language, and $M = |L|$. We normalize weights due to its value range.

3.1.3. Stabilizing Gradients

However, due to the existence of weights, the difference of gradients calculated from two batches can reach several orders of magnitude. So for weights in each batch, the softmax function is used to make the gradient norm close to before.

$$S(w_i) = \frac{e^{w_i}}{\sum_{k=1}^{n} e^{w_k}} \tag{5}$$

where e is the natural number, w_i means the target language's similarity of the i^{th} sample in a batch and n is the batch size. Furthermore, we put together the samples with larger differences in language similarity when constructing batches which makes the differences be more clearly reflected in training. Finally, the weight is simply multiplied with the original ASR loss.

$$\mathcal{L}_w = \sum_{i=1}^{n} S(w_i)\mathcal{L}_{jca}^{(i)} \tag{6}$$

where n denotes batch size and $\mathcal{L}_{jca}^{(i)}$ is the loss of the i^{th} sample.

3.2. Dynamic Curriculum Learning

The second proposed method is a warm-up strategy. The motivation of curriculum learning [22] is that the neural network can better utilize knowledge from easier examples rather than harder ones at the beginning. So the samples are ordered from easy to hard according to difficulty metrics for training. Inspired from the literature [23], we propose a dynamic curriculum learning method for low-resource ASR. Models are trained progressively instead of being fed with all samples.

3.2.1. Difficulty of Samples

For a training sample, lower loss means that the ASR model can better recognize it. So we adopt the loss of each sample as a measure of sample difficulty and use the frozen model to calculate the loss of all training samples after each training phase.

$$s(\mathbf{x}; \theta_t) = \mathcal{L}(\mathbf{x}; \theta_t) \tag{7}$$

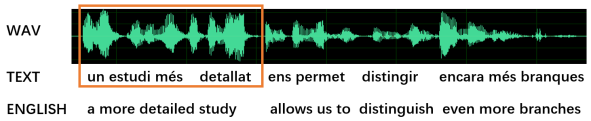

Fig. 2: Example of sub-sequence (Boxed part)

where $s(\mathbf{x};\theta_t)$ is the score of the sample $\mathbf{x}$ at phase t, and θ_t denotes the model parameters. On the other hand, another candidate for measuring difficulties is the negative accuracy $-a(\mathbf{x};\theta_t)$ of the attention output.

$$s(\mathbf{x};\theta_t) = -a(\mathbf{x};\theta_t) \tag{8}$$

Samples with smaller scores can be more difficult to be improved after several phases. Hereafter, we can also define the sample difficulties as the differences in scores of the same sample between adjacent phases. The sample difficulty based on score decreasing is expressed as

$$d(\mathbf{x};\theta_t) = -\frac{s(\mathbf{x};\theta_{t-1}) - s(\mathbf{x};\theta_t)}{s(\mathbf{x};\theta_{t-1})} \tag{9}$$

With this metric, samples with more reductions indicate that the model learns from them more efficiently. Therefore, they are more likely to be learned better in the next phase.

3.2.2. Progressive Learning

Generally, the model is weak in the early training stage, and then it can only learn well from the simple training samples and gradually learn to process the entire training set. Therefore, during the training process, we gradually increase the amount of training data to cover the entire training set. The proportion of training data is calculated as follows:

$$a(t) = \min\left(1,\ a_0 + \frac{\beta t}{T}(1 - a_0)\right) \tag{10}$$

where t means the t_{th} phase, a_0 means the initial ratio of data for training, β is the factor of data increment, and T means the total number of phases. Then for phase t, the $a(t) * |D_{train}|$ easiest samples are selected to train the model, where $|D_{train}|$ denotes the total size of the training set. Benefited from the progressive training, the newly-updated model can learn samples of appropriate difficulties.

3.3. Length Perturbation

Audios are resampled using different factors, and several additional copies of the data are created in speed perturbation [24]. Here we propose a new data augmentation strategy named "length perturbation" which can be well incorporated with the speed perturbation. Current end-to-end models learn the mapping of the whole sequence of input and output. However, there is a valid text sub-sequence corresponding to a piece of semantically segmented speech for the ASR task. Figure 2 shows an example of sub-sequence. The relationship between speech and text for "a more detailed study" in Catalan can be explicitly learned by models when we clip the sample to the boxed part for data augmentation. Then we can exploit knowledge from the sub-sequence of speech to improve the model's performance, especially in the case of scarcity of data. Subsequently, we first train a hybrid ASR system to get word boundaries for each utterance. Then we slice the utterances according to the word boundaries and create augmented samples. Consequently, explicit modeling of sub-sequences can help a lot for the ASR task.

4. EXPERIMENTS

4.1. Datasets

We consider five languages, including French (fr), Italian (it), Basque (eu), Portuguese (pt), and Catalan (ca) from the June 2020 (v5.1) release of CommonVoice Dataset[1] [25]. The training set of 1104 hours in total contains five languages. We rotate the role of the target 'low-resource language' so that only a 10-hour subset of the target language will be used. The official evaluation split of development and test sets is adopted for each language.

Table 1: WER (%) results of the proposed data weighing

Method	Catalan dev/test	Basque dev/test	French dev/test	Portuguese dev/test	Italian dev/test
Baseline	21.7/22.0	20.0/21.2	35.8/35.7	19.8/19.0	23.8/23.6
L.Post	21.5/21.7	19.4/19.9	34.6/34.6	19.6/18.6	22.7/22.6
L.Sim	21.5/21.6	19.3/19.9	34.6/34.7	19.4/18.5	22.7/22.7
U.Post	21.2/21.2	19.0/19.5	34.2/34.3	18.0/17.8	22.0/21.8
U.Sim	**20.3/20.4**	**18.5/19.0**	**34.1/34.2**	**17.6/17.0**	**21.2/21.1**

4.2. ASR Baseline

We follow the setup of the transformer model and the input in the literature [26]. The SpecAugment [27] is conducted on speech features, and the baseline implementation is from the ESPnet [28]. The modeling units are 500 byte pair encoding (BPE) units from the training set. The baseline performance is reported as the first line in Table 1 for all the five languages. "dev" and "test" means the word error rate (WER) on development and test sets, respectively.

4.3. Data Weighing Based on Language Similarity

The language classifier is trained to extract the language information from each utterance. We adopt the Time Delayed Neural Netowork (TDNN) structure from the literature [21]. The input of the model is the same as that in Section 4.2. The classifier is trained to identify which language the utterance comes from. We evaluate and compare the different strategies for data weighing. The system is firstly pretrained with the proposed data weighing and then finetuned on the target language.

The results for all the five languages are illustrated in Table 1. "Sim" means cosine similarity of embeddings, and "Post" means the posterior of the target language. "L" and "U" represent language-level and utterance-level, respectively. For example, "L.Post" means language-level data weighing with posterior strategy, and "U.Sim" indicates the utterance-level data weighing with similarity strategy. Compared with the language-level methods, the utterance-level methods have further improvements, indicating that the differences between utterances cannot be ignored. Due to the historical use and the existence of foreign words, some utterances in the non-target language have better benefits for the adaptation of the target language than that of other utterances. Also, the similarity based strategies achieve better performance than posteriors. The utterance level using the similarity strategy has achieved the best performance.

4.4. Dynamic Curriculum Learning

In all experiments, we set $a_0 = 0.2$, $\beta = 1.5$ in Eq. 10. And here, one phase corresponds to five epochs. After each phase, we infer on the whole training set to get the sample difficulties and reorganize the training set according to Eqs. 9 and 10.

As shown in Table 2, "DCL_L" means that the loss declination is considered as the metric of sample difficulty, and also "DCL_A" means the accuracy increase. The length normalized version is also introduced when we adopt loss value as the difficulty metric, which is denoted as "DCL_L*". For better comparison, the static curriculum learning (CL) is also conducted based on the literature [29].

[1] https://commonvoice.mozilla.org/en/datasets

Table 2: WER (%) of different curriculum learning methods

Methods	Catalan dev/test	Basque dev/test	French dev/test	Portuguese dev/test	Italian dev/test
Baseline	21.7/22.0	20.0/21.2	35.8/35.7	19.8/19.0	23.8/23.6
CL	22.0/22.2	20.8/21.7	35.9/35.8	19.9/19.0	23.8/23.7
DCL_A	20.9/21.1	18.8/19.2	34.0/34.1	18.6/17.4	23.0/22.8
DCL_L	21.0/21.0	18.4/19.0	33.6/33.6	18.5/17.4	22.6/22.4
DCL_L*	**20.4/20.6**	**17.4/18.3**	**33.0/33.1**	**17.8/16.7**	**21.8/21.6**

It is observed from Table 2 that the conventional curriculum learning does not work well for this low-resource scenario because it is static and the randomness is lost during training. With both proposed dynamic curriculum learning methods, for either loss-based or accuracy-based, the better performance is achieved. Also, further improvements are obtained when the normalized loss is adopted as the proposed dynamic curriculum learning.

4.5. Length Perturbation

Length perturbation requires the conversation time marked (CTM) output of training samples to segment them by word boundaries. We build the chain model of the CommonVoice recipe in Kaldi [30] using the 10-hour setup and align the training set of each language. The byte pair encoding units are adopted instead of phones because pronunciation lexicons of low-resource languages are hard to obtain.

Table 3: WER (%) Results of the proposed length perturbation

Methods	Catalan dev/test	Basque dev/test	French dev/test	Portuguese dev/test	Italian dev/test
Baseline	21.7/22.0	20.0/21.2	35.8/35.7	19.8/19.0	23.8/23.6
SP_3fold	20.2/20.6	**17.5**/18.0	32.5/**32.5**	**18.2/17.2**	21.3/21.3
LP_2fold	20.7/20.6	19.7/20.8	35.9/35.8	19.7/19.0	23.5/23.3
LP_3fold	20.1/20.1	18.7/20.6	34.1/34.1	18.8/18.2	22.1/22.7
LP_4fold	**20.1/19.8**	17.6/**17.9**	**32.4/32.5**	**18.2**/17.5	**20.7/20.6**
LP_5fold	20.2/20.0	18.1/19.1	33.2/33.3	18.7/18.0	21.2/21.0
SP+LP	**18.7/18.8**	**16.8/17.2**	**31.4/31.3**	**17.4/16.3**	**20.7/20.3**

Similar to the speed perturbation implementation in Kaldi, we perturb the training data with the proposed length perturbation using several different augmentation factors. We first select the starting point with a random word for each utterance and cut out the part of the text and the corresponding speech segment. By the use of factor **k**, we perturb data by $\frac{t}{k}, t \in \{1, 2, 3, ..., k\}$. Then we tried different factors for the proposed length perturbation, and results are shown in Table 3. In this table, "LP" and "SP" mean length perturbation and speed perturbation respectively. For speed perturbation, we use the broadly adopted configuration for speed factors 0.9, 1.0, and 1.1. It is found that the performance gets better while **k** is increased, and the best one is captured when $k = 4$. In comparison to the normal speed perturbation, the best length perturbation configuration performs better in most testing sets. More importantly, as shown the last line of Table 3, these two perturbation methods can be further combined to get much better performance.

4.6. Evaluation of Integrated Training Strategy

We evaluate and explore the integration of the proposed methods, including data weighing, dynamic curriculum learning, and length perturbation, and the results are shown in Table 4.

The last three lines show the results of our integrated methods. It is revealed that the proposed approaches are complementary with each other, and all the three methods can be combined into an entire training strategy to obtain the best system performance. There is a consistently relative 10% to 15% WER reduction compared to the system with speed perturbation. Compared to the baseline multilingual PT+FT, our final integrated data usage strategy incorporated with speed perturbation has a relative 15% to 25% WER reduction.

Table 4: WER (%) results of integrated training strategies for all five langauges. **M0**: Baseline. **M1**: **M0** + Speed perturbation. **M2**: **M1** + Length perturbation. **M3**: **M2** + Data weighing. **M4 (Final integrated strategy)**: **M3** + Dynamic curriculum learning.

Methods	Catalan dev/test	Basque dev/test	French dev/test	Portuguese dev/test	Italian dev/test
M0	21.7/22.0	20.0/21.2	35.8/35.7	19.8/19.0	23.8/23.6
M1	20.2/20.6	17.5/18.0	32.5/32.5	18.2/17.2	21.3/21.3
M2	18.7/18.8	16.8/17.2	31.4/31.3	17.4/16.3	20.7/20.3
M3	18.0/18.1	16.0/16.7	30.8/30.7	17.0/15.9	20.0/19.8
M4	**17.7/17.6**	**15.0/16.0**	**30.5/30.4**	**16.2/15.0**	**18.9/18.7**

4.7. Evaluation on Non Indo-European Lanugages

We adopt five Indo-European languages except Basque for the basic setup in our experiments. The other non Indo-European languages. Tatar (tt), Kabyle (kab) and Kinyarwanda (rw) are adopted as target languages to further evaluate our proposed approach. We use a 10-hour subset of the training set from each language, respectively. Most setups are the same as the basic experiments. While the model is first pretrained on 1104 hours of the full validated training set from five languages (fr, it, eu, pt, ca). Then we finetune the model on the target language (one of tt, kab, and rw), respectively. We replace the output layer with a new randomly initialized layer for each target language due to different modeling units.

Table 5: **WER (%) results of integrated data usage strategies for non Indo-European langauges.** The methods M0, M1, M2, M3, M4 are the same as those illustrated in Table 4.

Methods	Tatar dev/test	Kabyle dev/test	Kinyarwanda dev/test
M0	26.6/27.1	53.4/53.1	48.3/48.5
M1	23.3/23.9	51.0/51.7	45.7/46.0
M2	18.5/18.7	43.0/42.9	42.6/42.7
M3	17.8/18.1	42.5/42.3	41.5/41.6
M4	**16.2/16.2**	**40.9/40.8**	**37.4/37.7**

The performance of the proposed method is shown in Table 5. The absolute ASR performance in Kabyle and Kinyarwanda is not as good as Indo-European languages. It shows that the observation and conclusion are consistent as those in Table 4, and all the proposed methods still work well on non Indo-European languages. The proposed entire integrated data usage strategy can obtain a large improvement compared to the baseline multilingual PT+FT.

5. CONCLUSIONS

In this paper, we propose methods to explore effective data utilization for low-resource speech recognition. We use the language similarity for data weighing, dynamic curriculum learning for data allocation, and length perturbation for data augmentation. The experimental results show that on the CommonVoice dataset, all the three methods achieves better performance compared with the commonly used PT+FT baseline. The integrated proposed strategies achieve a 15-25% reduction in relative WER on different target languages.

6. ACKNOWLEDGEMENTS

This work was supported by the China NSFC projects (No. 62122050 and No. 62071288), and Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0102).

7. REFERENCES

[1] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *ICASSP*. IEEE, 2012, pp. 4269–4272.

[2] Y. Qian, K. Yu, and J. Liu, "Combination of data borrowing strategies for low-resource lvcsr," in *ASRU*. IEEE, 2013, pp. 404–409.

[3] Y. Qian, J. Xu, D. Povey, and J. Liu, "Strategies for using mlp based features with limited target-language training data," in *ASRU*. IEEE, 2011, pp. 354–358.

[4] S. Tong, P. N. Garner, and H. Bourlard, "Cross-lingual adaptation of a CTC-based multilingual acoustic model," *Speech Communication*, vol. 104, pp. 39–46, 2018.

[5] Y. Qian and J. Liu, "Articulatory feature based multilingual mlps for low-resource speech recognition," in *Interspeech*, 2012.

[6] S. Li, C. Ding, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "End-to-end articulatory attribute modeling for low-resource multilingual speech recognition." in *Interspeech*, 2019, pp. 2145–2149.

[7] S. Khare, A. Mittal, A. Diwan, S. Sarawagi, P. Jyothi, and S. Bharadwaj, "Low Resource ASR: The Surprising Effectiveness of High Resource Transliteration," in *Interspeech*, 2021, pp. 1529–1533.

[8] J. Xu, X. Tan, Y. Ren, T. Qin, J. Li, S. Zhao, and T.-Y. Liu, "LRSpeech: Extremely low-resource speech synthesis and recognition," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2802–2812.

[9] 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 *Proc. Interspeech 2020*, 2020, pp. 2817–2821.

[10] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," *Proc. Interspeech 2020*, pp. 1006–1010, 2020.

[11] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *arXiv preprint arXiv:2007.06028*, 2020.

[12] X. Song, G. Wang, Y. Huang, Z. Wu, D. Su, and H. Meng, "Speech-xlnet: Unsupervised acoustic model pretraining for self-attention networks," *Proc. Interspeech 2020*, pp. 3765–3769, 2020.

[13] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[15] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*, 2017, pp. 4835–4839.

[16] P. Shivakumar and P. Georgiou, "Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations," *Computer speech & language*, vol. 63, 2020.

[17] A. Kannan, A. Datta, T. 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*, 2019, pp. 2130–2134.

[18] S. Thomas, K. Audhkhasi, J. Cui, B. Kingsbury, and B. Ramabhadran, "Multilingual data selection for low resource speech recognition," *Interspeech*, pp. 3853–3857, 2016.

[19] A. Cutler, Y. Zhang, E. Chuangsuwanich, and J. R. Glass, "Language id-based training of multilingual stacked bottleneck features," in *Interspeech*, 2014.

[20] P. Dalsgaard and O. Andersen, "Identification of mono-and poly-phonemes using acoustic-phonetic features derived by a self-organising neural network." in *ICSLP*, 1992.

[21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[22] 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.

[23] C. Xu, B. Hu, Y. Jiang, K. Feng, Z. Wang, S. Huang, Q. Ju, T. Xiao, and J. Zhu, "Dynamic curriculum learning for low-resource neural machine translation," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 3977–3989.

[24] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Interspeech*, 2015, pp. 3586–3589.

[25] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[26] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs RNN in speech applications," in *ASRU*, 2019, pp. 449–456.

[27] 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," in *Interspeech*, 2019, pp. 2613–2617.

[28] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[29] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. ICML*, 2016.

[30] 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*, Dec. 2011.