Optimizing Data Usage for Low-Resource Speech Recognition

Yanmin Qian, *Senior Member, IEEE*, and Zhikai Zhou, *Student Member, IEEE*

***Abstract*—Automatic speech recognition has made huge progress recently. However, the current modeling strategy still suffers a large performance degradation when facing the low-resource languages with limited training data. In this paper, we propose a series of methods to optimize the data usage for low-resource speech recognition. Multilingual speech recognition helps a lot in low-resource scenarios. The correlation and similarity between languages are further exploited for multilingual pretraining in our work. We utilize the posterior of the target language extracted from a language classifier to perform data weighing on training samples, which assists the model in being more biased towards the target language during pretraining. Furthermore, dynamic curriculum learning for data allocation and length perturbation for data augmentation are also designed. All these three methods form the new strategy on optimized data usage for low-resource languages. We evaluate the proposed method using rich resource languages for pretraining (PT) and finetuning (FT) the model on the target language with limited data. Experimental results show that the proposed data usage method obtains a 15 to 25% relative word error rate reduction for different target languages compared with the commonly adopted multilingual PT+FT method on CommonVoice dataset. The same improvement and conclusion are also observed on Babel dataset with conversational telephone speech, and ∼40% relative character error rate reduction can be obtained for the target low-resource language.**

***Index Terms*—Low-resource speech recognition, curriculum learning, data augmentation, length perturbation.**

I. Introduction

AUTOMATIC speech recognition (ASR) is an entrance to human-machine interaction and attracts much attention in both research and industry communities. ASR benefits from a large quantity of parallel training data, i.e. speech with corresponding text labels, achieving human parity under ideal conditions. However, for low-resource languages, labeled data is much harder to be collected [1], [2]. While there are nearly 7,000 languages in the world, the vast majority of them suffer from the insufficiency of annotated data.

Manuscript received July 29, 2021; revised November 3, 2021 and December 13, 2021; accepted December 22, 2021. Date of publication January 5, 2022; date of current version January 21, 2022. This work was supported by China NSFC Projects under Grants 62122050 and 62071288. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sakriani Sakti. *(Corresponding author: Yanmin Qian.)*

The authors are with X-LANCE, Department of Computer Science and Engineering and MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yanminqian@gmail.com; zhikai.zhou@sjtu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2022.3140552

In order to solve the data scarcity problem in low-resource scenarios, many works are devoted to developing low-resource ASR approaches. One common method is to transfer knowledge from models trained on rich resource languages and adapt models to low-resource scenarios. Meanwhile, multilingual end-to-end ASR models avoid the pronunciation modeling, which is required in the traditional hybrid systems [3]–[5]. Inspired from the multilingual bottleneck feature [6]–[10] in hybrid ASR systems, the multi-headed output with shared encoder [11] established the basic architecture for the multilingual ASR model. LRSpeech [12] adopts text-to-speech (TTS) based data augmentation and dual transformation for both low-resource automatic speech recognition and speech synthesis. Meta-learning approaches want to solve the problem of fast adaptation on unseen data hoping to get a model that can be quickly adapted to low-resource languages [13]. Articulatory attributes are explored for modeling units because they are general for all human languages [14], [15]. Transliterations of different languages are adopted as data augmentation for multilingual models [16].

The exploitation of unlabeled data is also considered. Semi-supervised and self-supervised methods are proposed. Iterative pseudo-labeling and noisy student training distill the knowledge from the language model and data augmentation on additional unlabeled data [17]–[19]. They iteratively decode the model to predict hypotheses on unlabelled data with an external language model and train the model on augmented data with pseudo labels, which utilizes both unparalleled speech and text samples. Inspired from masked language models [20], masked acoustic models [21]–[23] are trained to predict the masked part of speech for self-learning. Then these models can be finetuned on a small amount of annotated data for low-resource speech recognition. Recently, wav2vec 2.0 [24] utilizes contrastive learning and masked acoustic models for self-learning. It takes only 10 minutes of annotated data to train an ASR model with decent performance. The integration of self-training and pretraining is investigated to push the limits of semi-supervised learning for ASR [25], [26], which achieves the state-of-the-art performance on the Librispeech [27] benchmark.

Although these methods achieve encouraging results for low-resource ASR, existing methods mainly focus on different training paradigms and the utilization of unlabeled data. On the other hand, weighing, scheduling, and training strategies for existing data are also essential perspectives but still haven't been well explored. This paper explores the advanced data usage strategies in detail for low-resource ASR.

First, for multilingual ASR or multilingual pretraining, prior works simply combine the data from different languages [3], [5], or sampling utterances according to a multinomial distribution to mitigate data imbalance [28], [29]. Some works do consider the relationship and correlation among languages [30], [31]. But they treat each language as a whole, and the main achievement is to reduce the training cost finally. Accordingly, we exploit similarities among languages in utterance level for better adaptation of low-resource ASR. Different strategies for computing similarities of training samples are explored.

Then, curriculum learning [32] is a kind of method for data allocation. Deep Speech 2 [33] proposes a static curriculum learning strategy named SortaGrad, which treats shorter utterances as simpler samples. However, it doesn't work well in our experiments due to the loss of randomness during training. Therefore, dynamic curriculum learning, which utilizes sample loss and its variation as the criterion for sample difficulty, is proposed in this paper. Such dynamic metrics are adopted as sample difficulties in our method. Model competence is also taken into consideration to be incorporated with sample difficulties for better optimization.

Furthermore, considering the monotonicity of speech recognition and the property that sequence-based models only have explicit modeling for the whole sentence, we propose length perturbation, which generates new samples based on utterance fragments for data augmentation. Hybrid ASR systems are used to segment the fragments from utterances. New created copies of training samples are generated for data augmentation.

More specifically, given the success of multilingual pretraining and finetuning for low-resource speech recognition, this paper further extends this framework by focusing on data usage and training strategy. The main contributions of this work are summarized as follows:

1) Compared with simply combining multilingual data for multilingual pretraining, the data weighing method based on utterance level language similarity is explored and proposed. Such similarities are exploited for better adaptation of low-resource ASR.
2) The novel dynamic curriculum learning method is designed to exploit the data scheduling scheme, which can make the model better optimized. We revise the order and training strategy of training samples, and both sample difficulties and model competence are taken into consideration.
3) A new data augmentation approach named length perturbation is developed for end-to-end ASR. It generates new samples based on utterance fragments, and can also be combined with existing data augmentation methods, i.e. speed perturbation [34] and SpecAugment [35].
4) Finally, an entire optimized data usage strategy based on all the above proposed methods is given. It is evaluated and justified to be effective on both CommonVoice [2] and Babel [1] datasets, and large improvement can be obtained for low-resource ASR.

The rest of the paper is organized as follows. Section II revisits the end-to-end ASR and the multilingual pretraining and finetuning framework for low-resource speech recognition at first. In Section III the proposed new data usage methods are described in detail, including data weighing, data allocation, and data augmentation. The experimental results and discussion are presented in Section IV, and finally, the conclusion is given in Section V.

II. Multilingual Pretraining and Finetuning for Low-Resource ASR

A. End-to-End ASR

The attention-based encoder-decoder (AED) transformer is adopted as the backbone in our ASR system. Transformer is a sequence-to-sequence (S2S) network [36].

The transformer model is trained under the joint connectionist temporal classification (CTC)/attention framework to improve robustness and achieve fast convergence [37], [38]. Note that CTC loss follows the output of the encoder. The CTC and S2S objectives are denoted by $\mathcal{L}_{ctc}$ and $\mathcal{L}_{s2s}$, and the loss function of the joint CTC-attention network is defined as:

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

A tunable coefficient $\lambda \in [0, 1]$ is applied to control the contribution of each loss.

1) Connectionist Temporal Classification (CTC): CTC merges the same consecutive tokens, and a special token **blank** is introduced in CTC to fill intermediate frames. Let $\mathbf{x}$ be the input feature sequence and $\mathbf{w}$ be the output word sequence. The probability $P(\mathbf{w}|\mathbf{x})$ is the summation of all possible CTC alignments.

$$P(\mathbf{w}|\mathbf{x}) = \sum_{\pi} P(\pi|\mathbf{x}) = \sum_{\pi} \prod_{t=1}^{T} P(\pi_t|\mathbf{x}) \tag{2}$$

where π represents possible alignments which can be mapped to $\mathbf{w}$ and $T = length(\pi) = length(\mathbf{x})$.

2) Attention-Based Encoder Decoder (AED): AED, also known as attention based sequence-to-sequence (S2S) model, forms the sequence labeling problem as a conditional language model problem. Note $\mathbf{w}$ and $\mathbf{x}$ be the output sequence and input feature. The criterion for AED, i.e. the probability $P(\mathbf{w}|\mathbf{x})$ is the product of the conditional probability of each single word by chain rule

$$P(\mathbf{w}|\mathbf{x}) = \prod_{i=1}^{N} P(w_i|\mathbf{x}, \mathbf{w}_{1:i-1}) \tag{3}$$

where $N = length(\mathbf{w})$. Joint CTC/attention decoding [39] is adopted to predict the output sequence, where S2S scores with CTC prefix scores are combined to make the final decision. We combine subword units [40] from all languages as the final units, and SpecAugment [35] is applied for all data in all our experiments.

B. Multilingual Pretraining and Finetuning

Multilingual pretraining is widely adopted for low-resource speech recognition [11]–[13]. Considering that many paired data

TABLE I
EXAMPLES OF WORD "PRONUNCIATION" FROM DIFFERENT LANGUAGES

Language	Word	IPA
Catalan	pronunciació	prununsiəsio
French	prononciation	prɔnɔ̃sjasjɔ̃
Italian	pronuncia	pronuntʃa
Portuguese	pronúncia	prunũsja
Basque	ahoskera	aʊʃkerə

IPA: International Phonetic Alphabet.

from rich-resource languages are already available, the E2E ASR model is first pretrained on several languages.

The main goal of multilingual pretraining is to share the data among multiple languages to learn the common knowledge across languages. Many current languages evolved from a common ancestor [41]. Therefore, it is natural that they share some common pronunciation and grammar among different languages. The pretrained model can learn common speech and language knowledge well based on such properties. Since larger models generally have more robust capabilities, a sufficient amount of data, even coming from different languages, allows us to avoid overfitting while using large models.

After the model is pretrained on rich resource languages, we finetune the ASR model on a low-resource language. Subword units from both rich and low-resource languages are adopted as modeling units. In this way, some common knowledge among different languages can be transferred to low-resource speech recognition by the pretrained parameters. After the finetuning, the optimized model can be applied for speech recognition on the target low-resource language.

III. OPTIMIZED DATA USAGE FOR LOW-RESOURCE SPEECH RECOGNITION

Models cannot be trained without data. However, the data usages of most works merely shuffle the order of data and train the model epoch by epoch. In the low-resource scenario, it is worth exploring how to make better use of data due to the data limitation for the target language. Therefore, in this paper, data weighing based on language similarity, data allocation based on dynamic curriculum learning, and data augmentation based on length perturbation are proposed to improve the performance of low-resource speech recognition.

A. Data Weighing Via Language Similarity

Multilingual pretraining simply combines data from different languages [42]–[44], and the work in [28], [45] samples utterances according to a multinomial distribution for multilingual training to avoid data imbalance. These approaches, however, fail to take advantage of the correlation and similarity among languages.

Previous work tried to use language level similarity for data selection to train the bottleneck feature in hybrid systems [30], [31]. In this paper, we further extend this idea to explore the utterance level language similarities and their benefits to low-resource language modeling with end-to-end ASR architecture.

Take a word as an example. As shown in Table I, for the word "pronunciation," Catalan has the very similar spelling and pronunciation with French, and the first four languages (Catalan, French, Italian, Portuguese) share the same prefix for the word "pronunciation". In contrast, Basque is totally different from others. Note that not all the words in the vocabulary have such properties, and the syntax in different languages also varies.

1) Data Weighing: In order to utilize the importance of samples in training, a common approach is used to divide the training data into subsets [31], which can be understood as straightforward data selection. However, this approach is not flexible enough in terms of implementation. Assuming that the training set is correctly labeled, any sampling may be useful for model learning but with different levels on the importance.

The data weighing method is proposed to utilize the similarity between the target language and non-target languages for ASR training. The purpose of using language similarity is to find data that is more similar to the target language in the multilingual dataset for better adaptation. In our approach, a language classifier is firstly constructed to obtain the similarity among languages in the utterance level. The posterior of the target language from the classifier can be considered as language similarity from the perspective of the model, which is then used as the weight of each utterance in multilingual pretraining.

Fig. 1 shows the whole procedure of the proposed method. The posteriors or similarities of the target language from the classifier are extracted as weights of samples. Then losses are multiplied with weights to make the model pay more attention to utterances with higher similarities.

2) Weights Computation: Equation 4 adopts the posterior from the TDNN based language classifier as sample weights w_i.

$$w_i = P(y = l|x_i) \tag{4}$$

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

The posteriors from the softmax layer are sometimes too close to one-hot vectors such that the weights of the non-target languages are too small. Inspired from speaker verification task [46], we also try to extract the embeddings from hidden layers. We average embeddings from each language as the language identifier embedding center, and compute the cosine similarity as weights w_i between language embedding center and utterance embeddings as follows.

$$cos\_sim(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||\,||\mathbf{b}||} \tag{5}$$

$$w_i = \frac{1 + cos\_sim(s_i, \frac{\sum_{k=1}^{|L|} s_k}{|L|})}{2} \tag{6}$$

where $cos\_sim(\mathbf{a}, \mathbf{b})$ is the cosine similarity between vector $\mathbf{a}$ and $\mathbf{b}$, s_i is the embedding of sample i. $s_k \in L$ where L is the set of target language. Since the value range of cosine similarity is [-1.0, 1.0], we normalize it to [0.0, 1.0].

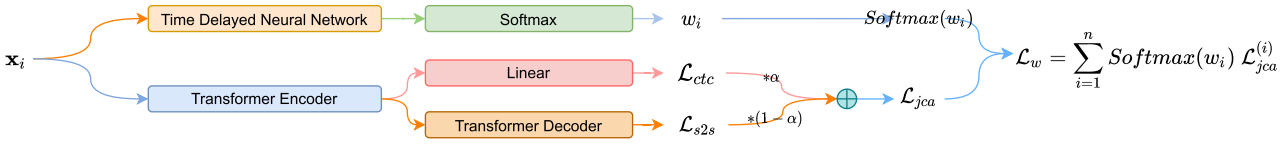

Fig. 1. The proposed data weighing procedure.

3) Stabilizing Weights: However, the vanilla weighing scheme leads to unstable training gradients, resulting in worse performance in our preliminary experiments. For example, due to the presence of weights, one batch has all weights of 0.1 and the other one has all 1.0, and the gradient norms of two batches can differ so much. So for weights in each batch, we transform them by the softmax function.

Furthermore, since similarities are scaled by the softmax function, we put together samples with larger differences in language similarity when constructing batches, which makes differences be more clearly represented in training in case the batch being full of utterances coming from the target language. The weight is multiplied with the original ASR loss to form the new loss function $\mathcal{L}_w$ for per batch on the E2E-ASR model training.

$$\mathcal{L}_w = \sum_{i=1}^{n} Softmax(w_i)\mathcal{L}_{jca}^{(i)} \tag{7}$$

where n is the batch size and $\mathcal{L}_{jca}^{(i)}$ means the joint CTC/attention ASR loss of the i^{th} utterance. Based on softmax, we can keep the gradient norm close to the original but prefer different samples based on the language weight.

B. Dynamic Curriculum Learning

The second proposed approach to optimize data usage is dynamic data allocation during training. Curriculum learning (CL) was first introduced in [32]. The motivation of CL is that the neural network can explore harder samples effectively by utilizing the prior knowledge learned from easier samples. So the samples are reordered from easy to hard in the training phase.

Inspired by [47], we propose a dynamic curriculum learning method for low-resource ASR. The order of training samples is determined dynamically rather than a static ordering. Furthermore, the model's competence is taken into consideration: models need to be trained progressively instead of being fed with all samples at once, regardless of their difficulties.

1) Problem Definition: Let $\mathbf{D}$ be the training dataset and $|\mathbf{D}|$ be the corpus size. Denote $\mathbf{x}$ as the training sample in $\mathbf{D}$. The training objective for ASR systems is to seek the optimal parameter θ minimizing the loss function $\mathcal{L}$ on the training set $\mathbf{D}$.

$$\theta = \arg\min_{\theta} \sum_{\mathbf{x}\in\mathbf{D}} \mathcal{L}(\mathbf{x};\theta) \tag{8}$$

In order to learn better model parameters by dynamic curriculum learning, we decompose the whole training process into multiple phases $T = (t_0, t_1, t_2, \ldots, t_k)$. For each phase, the sub-optimal process can be viewed as

$$\theta_{t+1} = \arg\min_{\theta_t} \sum_{\mathbf{x}\in\mathbf{D}_t} \mathcal{L}(\mathbf{x};\theta_t) \tag{9}$$

where θ_t means model parameters at phase t. $\mathbf{D}_t$ is the subset of the training set at phase t. Here a phase t can be a fixed number of epochs.

Previous works such as SortaGrad [33] enforces the static scoring strategy, which pre-determines data at each phase before training. However, the static order brings loss of randomness. We will further discuss such problem in Section IV-D.

2) Difficulty of Samples: For a training sample, a lower loss or a higher accuracy means the ASR model recognizes it better. Therefore, a simple way is to use the loss of each sample $\mathcal{L}(\mathbf{x};\theta^t)$ as the measurement of difficulty. To this end, we compute the score of all training samples with a frozen model after each

Algorithm 1: Dynamic Curriculum learning for low-resource ASR

```
Load the training dataset D;
Initialize score randomly.;
for each t in phases T do
    // one phase means k epochs
    Sort the training data in D in ascending order of score;
    D' := The first a(t) * |D| samples in D;
    for each i in epochs k do
        for each i in minibatches of D' do
            Feed minibatch i into the model and perform gradient descent;
        end
    end
    Fix model parameters θ;
    for each i in minibatches of whole training set D do
        Feed minibatch i into the model and update the score;
    end
end
while model is not converged do
    Shuffle the training data D randomly and divide them into minibatches;
    for each i in minibatches of the whole training set D do
        Feed minibatch i into the model and perform gradient descent;
    end
end
```

training phase.

$$s(\mathbf{x};\theta^t) = \mathcal{L}(\mathbf{x};\theta^t) \tag{10}$$

where $s(\mathbf{x};\theta^t)$ is the score of the sample $\mathbf{x}$ at phase t, and θ^t denotes the model parameters at phase t. Here, one phase can be a fixed number of epochs. Meanwhile, the loss value of longer utterances are usually bigger than shorter ones, so the length normalized loss can also be a candidate for measuring the difficulty of the sample.

$$s(\mathbf{x};\theta^t) = \frac{\mathcal{L}(\mathbf{x};\theta^t)}{T} \tag{11}$$

where T is the length of the output sequence. For the sequence-to-sequence model, the accuracy $a(\mathbf{x};\theta^t)$ of the attention output can also be adopted for measuring the difficulty of the sample. A higher accuracy means the sample is simpler.

$$s(\mathbf{x};\theta^t) = -a(\mathbf{x};\theta^t) \tag{12}$$

Since the model is updated in the training phase, a sample's loss may decrease rapidly after some epochs, and samples with smaller losses may be harder for improvement in training. So we can also define the CL score as the loss difference on sample between adjacent phases. The decline-based sample difficulty d is measured as

$$d(\mathbf{x};\theta^t) = -\frac{s(\mathbf{x};\theta^{t-1}) - s(\mathbf{x};\theta^t)}{s(\mathbf{x};\theta^{t-1})} \tag{13}$$

Also, the increment of accuracy can be the difficulty metric of the dynamic CL. With these metrics, samples with lower scores indicate the model learns them more effectively. So they are more likely to be learned better in the next phase.

3) Progressive Learning: The model is assumed to only learn well from the easiest training samples due to its weak capability at the early training stage and then gradually learn to handle the entire training set. Therefore, during the training process, we progressively increase the number of training samples until they cover the whole training set. The ratio a of training data in each phase is computed as follows:

$$a(t) = \min\left(1,\ a_0 + \frac{\beta t}{T}(1 - a_0)\right) \tag{14}$$

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 for dynamic curriculum learning. 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 with appropriate difficulties.

Algorithm 1 describes the entire strategy for the proposed dynamic curriculum learning. More specifically, we sort the training set by the scores computed in the last phases and use a subset of the first $a(t) * |D_{train}|$ samples. After that, the model is further trained with the random order over the whole set.

C. Length Perturbation

Speed perturbation [34] is an effective and commonly used method for data augmentation. The audios are resampled by different factors, and several additional copies of the data are created. Here we propose a new data augmentation strategy named length perturbation. The basic idea of length perturbation is similar to the recently proposed sub-sequence sampling in [48], [49], while derived from a different perspective.

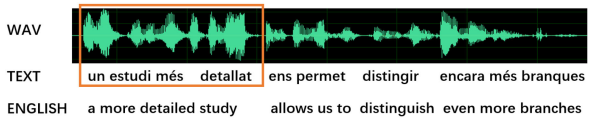

Fig. 2. Example of sub-sequence (Boxed part).

Current end-to-end AED models view the entire speech sequence as a whole. They are different from traditional hybrid acoustic models that classify a local context of speech features (usually one frame or several spliced frames). Furthermore, the ASR task is monotonic since there is a valid text sequence corresponding to a piece of semantically segmented speech. Depending on such property, we can exploit knowledge from the sub-sequence of speech to further improve the performance, especially when we do not have much data.

Fig. 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. For sequence-to-sequence models, the model only learns the correspondence of whole sentences. But for the ASR task, the mapping between some sub-sequences is also available. Such a relationship can be learned when there is a lot of data, but it is not available in low-resource scenarios.

Algorithm 2 describes the whole procedure. First, we get word boundaries according to alignments from the hybrid ASR system. Then we augment data based on a given factor k ($0 < k < 1$), where k is the factor to control the length of the new sequence compared to the origin sequence. Like speed perturbation, we augment the whole training set with a given factor k and generate a new copy of the training set with length factor k of the original utterance. We can augment data with different factors to generate more data with different lengths.

IV. Experiments

A. Experimental Setup

The CommonVoice Dataset[1] [2] is a massively multilingual corpus of transcribed speech. The contents of the corpus are mainly from Wikipedia articles. We utilize five languages in our experiments, including French (fr), Italian (it), Basque (eu), Portuguese (pt), and Catalan (ca). For the traditional approach, these five languages are pooled together for multilingual pretraining, and then the target language is used in finetuning. Table II shows details of languages in CommonVoice. The total speech duration of datasets ranges from 48 hours to 554 hours. We use the June 2020 (v5.1) release of CommonVoice. Note that only part of the dataset is officially validated. We adopt the full validated training set of 1104 hours in total for all five languages. For

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

Algorithm 2: Length Perturbation for low-resource ASR

Load the hybrid ASR system $\mathbf{M}$;
Load the training dataset $\mathbf{D}$;
Word boundary dict $\mathbf{W}$;
for *each sample utt,*$\mathbf{x}$,$\mathbf{y}$ *in* $\mathbf{D}$ **do**
  //Get the Conversation Time Marked (CTM) output
  $\mathbf{C} = Align(\mathbf{M}, \mathbf{x}, \mathbf{y})$;
  $\mathbf{W}[utt] = \{\}$;
  for *each (utt,s_t, e_t, word) in* $\mathbf{C}$ **do**
    // s_t and e_t mean start time and end time of the word, respectively
    Add {(s_t,e_t, word)} to $\mathbf{W}[utt]$
  end
end
Given perturb factors $\mathbf{F}$;
for *each factor* $\mathbf{f}$ *in* $\mathbf{F}$ **do**
  for *each sample utt,* $\mathbf{x}$, $\mathbf{y}$ *in* $\mathbf{D}$ **do**
    // The unit of length(y) is the number of words.
    $len := length(\mathbf{y})$;
    $newlen := len * \mathbf{f}$;
    $index := random\_int(0, len - newlen)$;
    $new\_start := \mathbf{W}[utt][index][0]$;
    $new\_end := \mathbf{W}[utt][index + newlen][1]$;
    // clip $\mathbf{x}$ by Sox
    $new\_\mathbf{x} := \mathbf{x}[new\_start : new\_end]$;
    $new\_\mathbf{y} := \mathbf{y}[index : index + newlen]$;
    Add new sample to $\mathbf{D_f}$;
  end
  $\mathbf{D}' := \mathbf{D}' \cup \mathbf{D_f}$;
end
Output perturbed dataset $\mathbf{D}'$

TABLE II
DETAILS OF FIVE LANGUAGES USED IN THE EXPERIMENTS FROM THE COMMONVOICE

Language	#Spk.	#Utt.	Duration
Catalan	4,742	317,693	488 hr
Basque	834	61,426	88 hr
Portuguese	717	39,072	48 hr
Italian	4,976	83,407	130 hr
French	11,381	412,332	554 hr

each language, we rotate the role of the target 'low-resource language'. We use a 10-hour subset from the target language and the full training set of the other four languages for pretraining, and the same 10-hour subset for finetuning. We evaluate our model on the official evaluation split of development and test sets for each language.

B. Baseline System

The input of the model is an 80-dimensional log Mel-filterbank with a 25 ms window length computed every 10 ms and 3 dimensional pitch features. The baseline implementation is from the ESPnet [50]. We adopt a transformer with 12 layers of encoder and 6 layers of decoder. Each layer is a transformer block with 4 heads of 64-dimensional self-attention layer and a 2048-dimensional feed-forward layer. Dropout is set to 0.1 after the feed-forward layer for each block. For the multi-task learning (MTL) in the joint CTC-attention optimization, the weights for CTC and attention loss are set to 0.3 and 0.7, respectively. The modeling units are 500 byte pair encoding (BPE) units trained from the whole multilingual training set consisting all five languages. SpecAugment [35] has been commonly used in ASR now, and it was also applied on log Mel-filterbank features in default in our experiments.

As mentioned in Section II, we first pretrain the model on the combination of five languages until convergence. Then we directly transfer all parameters for the target language ASR model, and only data from the target language is used for finetuning. The performance of this baseline system is shown as the first line in Table IV for all five languages.

TABLE III
WER (%) RESULTS ON THE SETUP WITH THE CATALAN AS THE TARGET LANGUAGE AND OTHERS ARE NON-TARGET LANGUAGES, AND ONE NON-TARGET LANGUAGE IS ABSENCE IN ROTATION FOR EACH SYSTEM WE HAVE 40 HOURS FOR EACH NON-TARGET LANGUAGE AND 10 HOURS FOR THE TARGET LANGUAGE CATALAN

Model	Duration	Dev	Test
All Languages	170 hr	25.1	25.2
W/O Basque	130 hr	**24.8**	**25.0**
W/O French	130 hr	25.3	25.6
W/O Italian	130 hr	26.4	26.7
W/O Portuguese	130 hr	25.4	25.8

TABLE IV
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**

C. Evaluation on the Proposed Data Weighing

In order to get the utterance level language similarity between the target language and other languages, a language classifier is trained to get the posterior from each utterance. We adopt the Time Delayed Neural Netowork (TDNN) structure from the literature [46]. We keep the same structure, except that the categories have been changed from speakers to languages and the hidden size has been adjusted to 256 to avoid overfitting. The input of the model follows the setup of the ASR model in Section IV-B. The classifier is trained to identify in which language the utterance is speaking. Because we need the language classifier to predict the language posterior on the ASR training set. Only 10 hours from each language is adopted to train the TDNN classifier regardless of the target language.

Firstly, we conduct preliminary experiments to show the importance of similarity among languages. We use Catalan as the target language, and remove different non-target languages in rotation in order to explore the impact of the absence of non-target languages on the performance of the model. To mitigate

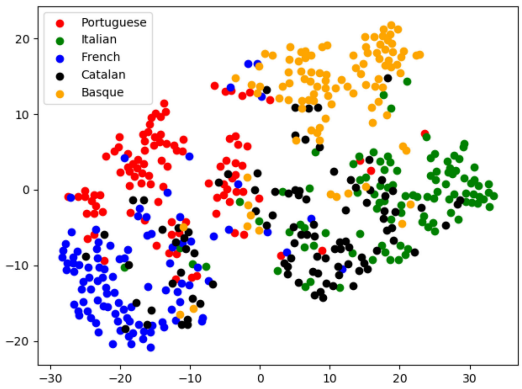

Fig. 3. T-SNE Visualization of language embeddings from the language classifier.

the impact of data volume, we reduce the amount of training data to 40 hours for each non-target language in these experiments, the same as Portuguese, which has the least amount of data.

Table III shows WER results comparison on the target Catalan of ASR systems trained from different non-target languages. The system without Basque achieves the best performance. It is even better than the system with all languages which has more data for training. Compared with the system without Italian, the system without Basque has a relative 6.4% WER improvement. This also indicates that Basque is not important to Catalan as a non-target language and even plays a counterproductive role. In contrast, Italian is probably the language that is the most similar and important to Catalan.

We also display the T-SNE visualization of embeddings from the language classifier for all five languages in Fig. 3. Points representing Catalan (black) are indeed most close to points of Italian (green). Part of them are also close to Portuguese (red) and French (blue). But most Basque (orange) points have an obvious distance from black points. Both the results in Table III and Fig. 3 show that the language similarity and acoustic distribution are important for the appropriate data usage in the multilingual modeling.

Finally we evaluate and compare the different strategies for data weighing. The system is firstly pretrained with the proposed data weighing, and then is finetuned on the target language. For comparison, we also attempt to scale all utterance with the same value for each language based on average LID scores, which can be named as language-level data weighing. The results for all five languages are illustrated in Table IV. In Table IV "Sim" means the similarity of embeddings from the language classifier between each utterance and the averaged language embedding center of the target language, and "Post" means the posterior of the target language from the language classifier for each utterance. "L" and "U" represent language-level data weighing and utterance-level data weighing, 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.

From Table IV, we can observe that all data weighing methods consistently outperform the baseline for all languages. The utterance-level approaches obtain further improvement compared with language-level ones. This is reasonable that the differences between sentences for adaptation cannot be ignored. Due to historical usage and the presence of foreign words, some sentences from non-target languages have a better gain for the target language adaptation. The utterance-level using similarity strategy achieves the best performance position, and it is obviously better than the others.

TABLE V
WER (%) COMPARISON OF DIFFERENT CURRICULUM LEARNING METHODS. CL: CONVENTIONAL STATIC CURRICULUM LEARNING USING UTTERANCE LENGTH. **DCL_A**: DYNAMIC CURRICULUM LEARNING USING TRAINING ACCURACY. **DCL_L**: DYNAMIC CURRICULUM LEARNING USING TRAINING LOSS. **DCL_L***: DYNAMIC CURRICULUM LEARNING USING LENGTH NORMALIZED TRAINING LOSS

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**

D. Evaluation on the Proposed Dynamic Curriculum Learning

The proposed dynamic curriculum learning is evaluated here. In all experiments, we set $a_0 = 0.2$, $\beta = 1.5$ in Eq. 14, and a phase t corresponds to five epochs. For the first phase, we use the random scores for initialization. After each phase, we first forward the whole training set to get the loss or accuracy and reorganize the training set according to Eq. 13 and Eq. 14. Because we need to forward throughout the whole training set, we adopt data-parallel for distributed inference. Each task only computes the scores and reorder the training samples on their own part.

As shown in Table V, "DCL_L" means the loss declination is considered as the metric on difficulty for training samples, and "DCL_A" means the the increased accuracy of attention head is used for difficulty metric. When using the loss value, longer utterances are larger than shorter ones. So another length normalized version is also introduced when we adopt loss value as the difficulty metric, which is denoted as "DCL_L*". For sequence-to-sequence tasks, one commonly used conventional curriculum learning is to treat shorter utterances as easier samples, and model is optimized according to the utterance length [33]. This is a static curriculum learning strategy, also named as SortaGrad in [33]. For better comparison, this static curriculum learning is also conducted, and to be compared to our proposed dynamic curriculum learning.

It is observed that the conventional curriculum learning does not work well for this low-resource scenario because its static and randomness is lost during training. Both proposed dynamic curriculum learning methods, either loss-based or accuracy-based, achieve obviously better performance compared to the baseline and conventional curriculum learning. In addition, further improvements can be obtained when the normalized loss is adopted as the proposed dynamic curriculum learning.

TABLE VI
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**

TABLE VII
WER (%) RESULTS OF INTEGRATED DATA USAGE STRATEGIES FOR ALL FIVE LANGUAGES. 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**

E. Evaluation on the Proposed Length Perturbation

Length perturbation requires the conversation time marked (CTM) output of training samples to segment them by accurate word boundaries. We follow the CommonVoice recipe in Kaldi [51] to build a hybrid ASR model and align the training set of each language. The chain model is an 8-layer TDNN with 768 hidden dimensions. The input of the model is composed of the 40-dimensional mel-frequency cepstrum (MFCC) with a 25 ms window length computed every 10 ms and a 100-dimensional i-vector for speaker adaptation. The subword units BPE are adopted instead of phones because a pronunciation lexicon of a low-resource language is usually hard to be obtained.

Similar to the speed perturbation implementation in Kaldi, we perturb the training data with proposed length perturbation using several different augmentation factors. The newly created copies of training data have different factors of original text lengths. Different factors are chosen according to how many folds do we need for data augmentation. If we need k folds of data, we will perturb data by factor $\frac{t}{k}$, $t \in \{1, 2, 3, \ldots, k\}$. Shown as Table VI, for example, "LP_2fold" means we perturb data by factor $\frac{t}{2}$, $t \in \{1, 2\}$. Factor 1.0 means the original training data. The newly created copies of training data have the factor of 0.5,1.0 for utterance length. We first select the starting point with a random word for each utterance and cut out a part of the text and corresponding speech segment. After that, SoX [52] is adopted to clip the audio into new samples according to the CTM output. We do the perturbation at both multilingual pretraining and finetuning stages.

We tried different factors $\mathbf{K}$ for the proposed length perturbation, and the results are shown in Table VI. "LP" and "SP" mean length perturbation and speed perturbation respectively. "#fold" means we perturb the training set with a different number of copies, where "3fold" means two copies of augmented data are created plus the original training set. For speed perturbation, we use the broadly adopted configuration for speed factors 0.9, 1.0, and 1.1. It is observed that compared with the baseline, all perturbation approaches obtain obvious WER reduction. When implementing the proposed length perturbation with different factors $\mathbf{K}$, better performance is obtained when $\mathbf{K}$ is increased and the system achieves the best position when $k = 4$. Doing the comparison between the proposed length perturbation and the normal speech perturbation, the best length perturbation configuration sightly outperforms in most testing set. In fact, shown as the last line of Table VI, these two perturbation methods can be combined to get a further improved performance.

F. Evaluation on the Integrated Data Usage Strategy

Finally we evaluate and explore the integration of the above proposed methods, including data weighing, dynamic curriculum learning and length perturbation, and the results are shown in Table VII. According to the results above, utterance-level similarity based data weighing, dynamic curriculum learning using length normalized loss, and length perturbation with factor $k = 4$ are chosen for our integrated evaluation.

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

G. Evaluation on Non Indo-European Languages

For the basic setup in our experiments, we adopt five languages written in Latin characters and all of them are Indo-European languages except Basque. Therefore, we attempt to further extend and evaluate our proposed approach to other non Indo-European languages, and Tatar (tt), Kabyle (kab) and Kinyarwanda (rw) are adopted as target languages. Following the previous setup, we use a 10-hour subset of the training set from each language, respectively. Then we evaluate the model on the official evaluation splits.

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). We replace the output layer with a new randomly initialized layer for each target language due to different character sets.

The results comparison of the proposed method is shown in Table VIII. Since languages are not Indo-European, the absolute ASR performance in Kabyle and Kinyarwanda is not as good as others. It shows that the observation and conclusion are consistent as those in Table VII, 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.

TABLE VIII
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 VII

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**

TABLE IX
CER (%) RESULTS OF SYSTEMS ON BABEL CONVERSATIONAL TELEPHONE SPEECH. B0: BASELINE. **B1**: **B0** + SPEED PERTURBATION. **B2**: **B1** + PROPOSED FINAL INTEGRATED DATA USAGE STRATEGY

Methods	Cantonese dev/test	Mongolian dev/test	Javanese dev/test
B0	69.2/70.5	89.7/92.9	84.7/84.4
B1	67.8/68.3	85.1/84.2	82.8/83.5
B2	**38.2/37.9**	**54.0/53.7**	**53.1/53.3**

H. Evaluation on Babel Conversational Telephone Speech

We also evaluate the proposed method on Babel corpus, which is conversational telephone speech data. Our multilingual model is firstly trained with nine languages - Amharic, Cantonese, Guarani, Javanese, Kurmanji-Kurdish, Mongolian, Pashto, Tamil, Vietnamese from the IARPA Babel program [1] and Somali from the IARPA MATERIAL program. Cantonese, Javanese, and Mongolian are selected as the target languages in our experiments. For each language, we have 10 hours for training, and 5 hours for development and test set, respectively.

The evaluation results are illustrated in Table IX, and the baseline is the normal multilingual pretraining + finetuning system. The proposed final integrated data usage strategy, including data weighing, dynamic curriculum learning and length perturbation, is applied. It is observed that the usual speed perturbation only can obtain very limited improvement on Babel, maybe due to the longer utterance and the uneven silence with pauses through segments in Babel conversational telephone speech corpus. In contrast, the proposed newly optimized data usage strategy achieves very large and consistent improvement for all the target low-resource languages, and relative 36% to 48% character error rate reduction is observed for all three target languages.

V. CONCLUSION

In this paper, we developed a series of methods and strategies to optimize the data usage for low-resource speech recognition. Three types of data usage methods are designed, including data weighing based on language similarities, data allocation based on dynamic curriculum learning, and data augmentation based on length perturbation. All these three methods can effectively make better use of limited training data and significantly improve the system performance for low-resource speech recognition. Furthermore, all these three methods can be integrated into one entire date usage strategy to achieve better system performance. The proposed optimized data usage methods are evaluated on both the CommonVoice dataset and Babel conversational telephone speech dataset, and the experimental results show that all the proposed data usage methods can obtain a large improvement upon the commonly used multilingual PT+FT framework. In future work, the proposed methods can be further incorporated with other approaches such as semi-supervised learning [19], [29] and self-learning [28] for low-resource speech recognition.

ACKNOWLEDGMENT

The authors would like to thank Wei Wang, Yizhou Lu, and Wangyou Zhang for discussion, English proofreading, and editing.

REFERENCES

[1] M. Harper, "IARPA babel program," 2012. [Online]. Available: http://www.iarpa.gov/index.php/research-programs/babel
[2] R. Ardila *et al.*, "Common voice: A massively-multilingual speech corpus," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 4218–4222.
[3] S. Toshniwal *et al.*, "Multilingual speech recognition with a single end-to-end model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4904–4908.
[4] 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5621–5625.
[5] J. Cho *et al.*, "Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 521–527.
[6] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 4269–4272.
[7] Y. Qian and J. Liu, "Cross-lingual and ensemble MLPs strategies for low-resource speech recognition," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 2582–2585.
[8] Y. Qian, D. Povey, and J. Liu, "State-level data borrowing for low-resource speech recognition based on subspace GMMs," in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 553–556.
[9] Y. Qian, K. Yu, and J. Liu, "Combination of data borrowing strategies for low-resource LVCSR," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 404–409.
[10] Y. Qian, J. Xu, D. Povey, and J. Liu, "Strategies for using MLP based features with limited target-language training data," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011, pp. 354–358.
[11] S. Tong, P. N. Garner, and H. Bourlard, "Cross-lingual adaptation of a CTC-based multilingual acoustic model," *Speech Commun.*, vol. 104, pp. 39–46, 2018.
[12] J. Xu *et al.*, "LRSpeech: Extremely low-resource speech synthesis and recognition," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2802–2812.
[13] J.-Y. Hsu, Y.-J. Chen, and H.-Y. Lee, "Meta learning for end-to-end low-resource speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7844–7848.
[14] Y. Qian and J. Liu, "Articulatory feature based multilingual MLPs for low-resource speech recognition," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 2602–2605.
[15] 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 *Proc. Interspeech*, 2019, pp. 2145–2149.
[16] S. Thomas, K. Audhkhasi, and B. Kingsbury, "Transliteration based data augmentation for training multilingual ASR acoustic models in low resource settings," in *Proc. Interspeech*, 2020, pp. 4736–4740.
[17] Y. Qian and J. Liu, "MLP-HMM two-stage unsupervised training for low-resource languages on conversational telephone speech recognition," in *Proc. Interspeech*, 2013, pp. 1816–1820.
[18] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," in *Proc. Interspeech*, 2020, pp. 1006–1010.

[19] D. S. Park *et al.*, "Improved noisy student training for automatic speech recognition," in *Proc. Interspeech*, 2020, pp. 2817–2821.
[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., Volume 1*, 2019, pp. 4171–4186.
[21] A. T. Liu, S.-W. Yang, P.-H. Chi, P.-C. Hsu, and H.-Y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6419–6423.
[22] X. Song, G. Wang, Y. Huang, Z. Wu, D. Su, and H. Meng, "Speech-XLNet: Unsupervised acoustic model pretraining for self-attention networks," in *Proc. Interspeech*, 2020, pp. 3765–3769.
[23] A. T. Liu, S.-W. Li, and H.-Y. Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2351–2366, 2021.
[24] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12449–12460, 2020.
[25] Q. Xu *et al.*, "Self-training and pre-training are complementary for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3030–3034.
[26] Y. Zhang *et al.*, "Pushing the limits of semi-supervised learning for automatic speech recognition," 2020, *arXiv:2010.10504*.
[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.
[28] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," 2020, *arXiv:2006.13979*.
[29] Z.-Q. Zhang, Y. Song, M.-H. Wu, X. Fang, and L.-R. Dai, "XLST: Cross-lingual self-training to learn multilingual representation for low resource speech recognition," 2020, *arXiv:2103.08207*.
[30] A. Cutler, Y. Zhang, E. Chuangsuwanich, and J. R. Glass, "Language id-based training of multilingual stacked bottleneck features," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1–5.
[31] S. Thomas, K. Audhkhasi, J. Cui, B. Kingsbury, and B. Ramabhadran, "Multilingual data selection for low resource speech recognition," in *Proc. Interspeech*, 2016, pp. 3853–3857.
[32] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.
[33] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 173–182.
[34] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3586–3589.
[35] D. S. Park *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.
[36] A. Vaswani *et al.*, "Attention is all you need," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5998–6008, 2017.
[37] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 4835–4839.
[38] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. Interspeech*, 2017, pp. 949–953.
[39] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2017, pp. 518–529.
[40] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 66–75.
[41] S. Tong, P. N. Garner, and H. Bourlard, "Multilingual training and cross-lingual adaptation on CTC-based acoustic model," 2017, *arXiv:1711.10025*.
[42] P. Shivakumar and P. Georgiou, "Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations," *Comput. Speech Lang.*, vol. 63, 2020, Art. no. 101077.
[43] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," 2018, *arXiv:1806.05059*.
[44] A. Kannan *et al.*, "Large-scale multilingual speech recognition with a streaming end-to-end model," in *Proc. Interspeech*, 2019, pp. 2130–2134.
[45] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 7059–7069, 2019.
[46] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5329–5333.
[47] C. Xu *et al.*, "Dynamic curriculum learning for low-resource neural machine translation," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 3977–3989.
[48] T.-S. Nguyen, S. Stueker, J. Niehues, and A. Waibel, "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7689–7693.
[49] T. K. Lam, M. Ohta, S. Schamoni, and S. Riezler, "On-the-fly aligned data augmentation for sequence-to-sequence ASR," in *Proc. Interspeech*, 2021, pp. 1299–1303.
[50] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.
[51] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011.
[52] SoX, "Audio manipulation tool," 2015. [Online]. Available: http://sox.sourceforge.net/

Yanmin Qian (Senior Member, IEEE) received the B.S. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2012. Since 2013, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is currently an Associate Professor. From 2015 to 2016, he was also an Associate Research with Speech Group, Cambridge University Engineering Department, Cambridge, U.K. His current research interests include acoustic and language modeling in speech recognition, speaker and language recognition, speech enhancement and separation, key word spotting, and multimedia signal processing.

Zhikai Zhou (Student Member, IEEE) received the B.Eng. degree from the Department of Software Engineering, Southeast University, Nanjing, China, in 2019. He is currently working toward the M.E. degree with the X-LANCE Lab., Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai, China, under the supervision of Yanmin Qian. His current research focuses on speech recognition.