Universal Cross-Lingual Data Generation for Low Resource ASR

Wei Wang, *Graduate Student Member, IEEE*, and Yanmin Qian, *Senior Member, IEEE*

***Abstract*—Significant advances in end-to-end (E2E) automatic speech recognition (ASR) have primarily been concentrated on languages rich in annotated data. Nevertheless, a large proportion of languages worldwide, which are typically low-resource, continue to pose significant challenges. To address this issue, this study presents a novel speech synthesis framework based on data splicing that leverages self-supervised learning (SSL) units from Hidden Unit BERT (HuBERT) as universal phonetic units. In our framework, the SSL phonetic units serve as crucial bridges between speech and text across different languages. By leveraging these units, we successfully splice speech fragments from high-resource languages into synthesized speech that maintains acoustic coherence with text from low-resource languages. To further enhance the practicality of the framework, we introduce a sampling strategy based on confidence scores assigned to the speech segments used in data splicing. The application of this confidence sampling strategy in data splicing significantly accelerates ASR model convergence and enhances overall ASR performance. Experimental results on the CommonVoice dataset show 25-35% relative improvement for four Indo-European languages and about 20% for Turkish using a 4-gram language model for rescoring, under a 10-hour low-resource setup. Furthermore, we showcase the scalability of our framework by incorporating a larger unsupervised speech corpus for generating speech fragments in data splicing, resulting in an additional 10% relative improvement.**

***Index Terms*—Low-resource speech recognition, text-to-seech, data splicing, self-supervised learning.**

I. Introduction

END-TO-END (E2E) models for automatic speech recognition (ASR) have attracted considerable attention in recent years due to their streamlined design and promising output [1], [2], [3], [4]. However, the effectiveness of E2E ASR models heavily relies on the availability of large quantities of transcribed audio data, which is often lacking in low-resource settings [5]. This scarcity of data poses a substantial challenge to deploying E2E ASR models for languages with limited resources. To address this challenge, researchers have proposed various solutions, including strategies such as multilingual transfer learning (MultiASR), multilingual meta-learning [6], [7], [8], [9], [10], [11], and self-supervised learning techniques [12], [13], [14], [15], [16], [17].

Both multilingual transfer learning and multilingual meta-learning are approaches that leverage labeled data to pre-train a foundational model using data from multiple languages. This pre-trained model serves as a starting point during the fine-tuning phase [6], [18], [19], [20], [21]. By constraining the parameter search space, these approaches facilitate faster convergence when working with low-resource languages [8], [22]. To further enhance the performance of low-resource ASR, [20] introduced an auxiliary speech-to-text translation task that converts labeled speech from a resource-rich language into text in a resource-poor language. Additionally, [21] proposed fine-tuning the base model's parameters using meta-learning techniques to enable rapid adaptation to various languages. However, it is important to note that both multilingual transfer learning and meta-learning require paired data for both the pre-training and fine-tuning stages.

Recently, the utilization of unlabeled data has emerged as a promising strategy, leading to the development and deployment of semi-supervised and self-supervised techniques. Two semi-supervised methods, namely iterative pseudo-labeling and noisy student training, leverage language model (LM) insights and data augmentation on extra unlabeled data [23], [24], [25]. These methods involve a repetitive process of decoding the model to generate hypotheses on unlabeled data, aided by an external language model. The model is then trained on augmented data using pseudo-labels, incorporating both unmatched speech and text samples. Simultaneously, self-supervised learning (SSL) has gained prominence for various tasks, leveraging readily available unpaired speech data to derive semantic information [12], [26], [27], [28]. Inspired by masked language models [29], masked acoustic models have been designed for self-learning, predicting the masked segment of speech [30], [31], [32]. These models can be fine-tuned with a modest quantity of annotated data in low-resource scenarios, resulting in competent ASR models [33]. For example, XLSR-53 [34] and XLS [35], pre-trained on 56 k and 500 k hours of speech data, respectively, have demonstrated notable performance across multiple languages, highlighting the efficacy of SSL models.

The aforementioned approaches rely on paired or unpaired real-world data from various languages, either pretraining a seed model for fine-tuning in low-resource target languages or

Manuscript received 30 May 2023; revised 15 October 2023 and 2 December 2023; accepted 8 December 2023. Date of publication 22 December 2023; date of current version 5 January 2024. This work was supported in part by China NSFC projects under Grants 62122050 and 62071288 and in part by the Shanghai Municipal Science and Technology Commission Project under Grant 2021SHZDZX0102. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Samuel Thomas. *(Corresponding author: Yanmin Qian.)*

The authors are with the Auditory Cognition and Computational Acoustics Lab, the Department of Computer Science and Engineering and the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wangwei.sjtu@sjtu.edu.cn; yanminqian@sjtu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3345150

iteratively optimizing the model with pseudo-labels. In contrast, this study addresses the data scarcity issue through an optimized text-to-speech (TTS) framework based on data splicing. Our framework generates additional paired data in low-resource languages by exploiting unpaired speech and text from rich-resource languages. The effectiveness of neural TTS models in improving ASR performance has been well-documented [36], [37], [38], [39], [40], [41], [42], [43]. For instance, previous studies have successfully applied neural TTS models to adapt the recurrent neural network transducer (RNN-T) [44] model from the source domain to the target domain [36], [37]. Similarly, researchers have utilized the machine speech chain framework to achieve mutual adaptation between TTS and ASR models from the audiobook to the presentation domains [39]. The work presented in [45] encompasses data augmentation techniques rooted in TTS methodologies, along with dual transformation operations, specifically tailored to improve performance in both automatic speech recognition and speech synthesis under low-resource conditions. To address the reduction in recognition accuracy for out-of-vocabulary (OOV) words, previous studies have trained ASR models on audio synthesized from text that includes OOV words [40], [41]. Additionally, researchers have improved the efficacy of a children's ASR system by enhancing the quality of synthesized children's speech using various filtering algorithms [43]. However, neural TTS models typically require a significant amount of high-quality data for effective training. Despite the reduction in the need for high-quality single-speaker paired data for these models [45], [46], [47], [48], synthesizing consistent speech for multiple speakers guided by noisy ASR data in low-resource conditions remains a challenging task. Moreover, the substantial computational costs associated with the training and inference stages of neural TTS models pose a significant barrier to training an ASR model using dynamically synthesized speech.

In [49], a TTS method based on word-level splicing data generation (SDG) was proposed. The method demonstrated favorable performance compared to neural TTS methods in text-only domain adaptation tasks in ASR, while maintaining minimal computational costs. The approach introduced in [49] establishes a correspondence between words and speech fragments using a tailored RNN-T model. This enables the association of text from the target domain with speech fragments from the source domain, which are then spliced to create coherent speech. It's highlighted in [49] that E2E models like RNN-T make decisions after processing segments of speech rather than frame-by-frame. Therefore, discontinuities at word transitions in spliced speech, while noticeable to humans, might not impact ASR models similarly. However, the word-level SDG presented in [49] is limited to monolingual scenarios, as universal word tokens across languages do not exist.

In response to this limitation, this study introduces a novel data splicing framework specifically designed for cross-lingual scenarios. The framework involves concatenating speech fragments from rich-resource languages to synthesize coherent speech that aligns with the text from a low-resource language. To achieve this, we leverage denoised clustering units (Hunits) extracted from the latent representations of a pretrained Hidden Unit BERT (HuBERT) [13] model as universal phonetic units that transcend language boundaries. We establish a correspondence, referred to as HuDict, between Hunit n-grams and speech fragments. Additionally, we develop a lightweight Grapheme-to-Hunit (G2H) conversion model. During the speech synthesis process, the G2H model maps text samples from the low-resource language to Hunits, which are then mapped to corresponding speech fragments using the HuDict. These speech fragments are concatenated to generate synthesized speech that acoustically aligns with the input text. A comprehensive overview of the proposed framework is presented in Fig. 1 and detailed in Section II-B.

The contributions of this study are as follows:

1) We propose a cross-lingual data splicing framework based on self-supervised learning (SSL) units that enables training of low-resource ASR model using on-the-fly synthesized speech.
2) We validate the feasibility of adopting HuBERT units (Hunits) as phonetic units by comparing them with phonemes in monolingual scenarios.
3) Experimental results conducted on multiple low-resource languages consistently demonstrate a reduction in word error rate (WER) by incorporating cross-lingually spliced data into the training of low-resource ASR models, showcasing the effectiveness of our approach.
4) The scalability of the proposed framework is validated by incorporating a larger unsupervised speech corpus as the source of speech fragments for data splicing.

II. Cross-Lingual Data Splicing With Self Supervised Phonetic Units

In this section, we provide a comprehensive description of the principles and design of speech synthesis based on data splicing. We begin by addressing the specific scenario of monolingual data splicing, where the target text and available speech fragments belong to the same language. This serves as a demonstration to highlight the challenges and limitations when applying the methodology in a cross-lingual context. The complexities of cross-lingual scenarios arise from the absence of a universally shared phonetic framework and the significant discrepancies in linguistic structures across different languages.

To navigate these challenges, we introduce a novel framework that enables cross-lingual data splicing using self-supervised phonetic units. These units offer a practical solution in cross-lingual scenarios, even though they may be less precise compared to phonemes within the same language, as universal phonemes do not exist.

Additionally, we detail an approach that incorporates confidence sampling to enhance the quality of synthesized speech. This technique plays a pivotal role in further boosting ASR performance, expediting model convergence, and substantially enhancing the overall efficacy of the proposed framework.

A. Data Splicing Within the Same Language

A feasible approach to speech synthesis through data splicing [50] typically involves the selection and concatenation of specific speech segments corresponding to phoneme sequences in the desired text. This approach, referred to as phoneme-guided

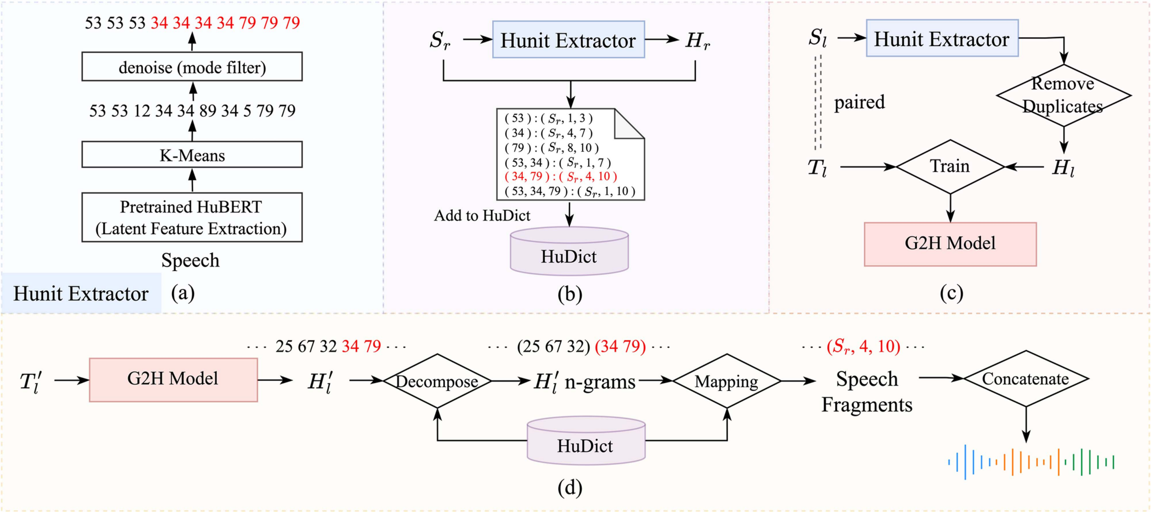

Fig. 1. Data splicing framework proposed in this study consists of the following steps. (a) An Hunit extractor takes the raw waveform as input and derives denoised clustering units, referred to as 'Hunits' from latent representations. (b) Given a source S_r, the Hunit extractor generates H_r and establishes associations between Hunit n-grams within H_r and their corresponding speech fragments in S_r. These mappings are then incorporated into the Hunit Dictionary (HuDict). (c) A small set of paired data, represented as $\{S_l, T_l\}$, are processed by the Hunit extractor. After removing any duplications, this data is used to train a Grapheme-to-Hunit (G2H) model. (d) During the synthesis stage, the G2H model transforms T_l' into H_l', which are then mapped to their corresponding speech segments using HuDict. The resulting speech fragments are concatenated to form the synthesized speech.

data splicing, relies on phonemes as the fundamental units of speech to identify the appropriate speech segments for generating synthetic speech. By utilizing phonemes, this approach offers a more extensive range of speech segment options and ensures a higher fidelity to the acoustic characteristics of the target text. An essential component of this technique is the construction of a phoneme dictionary, which involves mapping phonemes to their corresponding speech segments. This dictionary is built by employing forced alignment on a corpus of paired speech and text. Additionally, a language-specific lexicon is required to convert the text into a phoneme sequence during the forced alignment process.

While this approach has proven effective for data splicing in monolingual scenarios, its application to cross-lingual synthesis faces several challenges. One of the primary challenges is the absence of a universal phonetic system that encompasses all languages, making it difficult to map text from different languages to a sequence of universal phonemes. This limitation hinders the direct transferability of the approach across linguistic boundaries. Although the International Phonetic Alphabet (IPA) may seem like a potential solution, constructing an IPA table for each language is a manual and specialist-dependent process. Furthermore, while IPA is universal, it may not accurately capture subtle pronunciation differences influenced by specific linguistic variations. Another limiting factor is the requirement for paired speech-text corpora to build the phoneme dictionary. This constraint prevents the utilization of a significantly larger volume of unpaired speech data, thereby limiting the scope and scalability of the phoneme-guided data splicing approach in a cross-lingual setting.

B. Data Splicing Across Different Languages

To navigate the challenges inherent in cross-lingual data splicing, we have devised a novel strategy that relies on Hunits, which are phonetic units extracted from the pretrained self-supervised model HuBERT. Hunits serve as universal counterparts to language-specific phonemes and form the foundation of our cross-lingual approach, allowing us to overcome linguistic barriers in our work.

By leveraging a pretrained HuBERT model to extract Hunits from unpaired speech, we construct an Hunit dictionary, eliminating the requirement for large speech-text paired corpora that were previously essential for building a phoneme dictionary in the monolingual data splicing framework.

For converting text from different languages into Hunits, we employ a lightweight Grapheme-to-Hunits model, trained on a limited amount of supervised data for each target low-resource language. By establishing a shared phonetic unit representation, Hunits, we successfully create a connection between the speech and text modalities across diverse languages, enabling seamless integration through data splicing.

1) Framework Overview: Our proposed framework, depicted in Fig. 1, consists of three interconnected components, illustrated in Fig. 1(a), (b), and (c), with the audio synthesis process demonstrated in Fig. 1(d). The framework requires the following datasets:

i) D_r: Unlabelled speech samples extracted from the rich-resource language, with samples denoted as S_r.
ii) D_l: A limited quantity of speech-text pairs available in the low-resource language, consisting of pairs $\{S_l, T_l\}$.
iii) D_l': Text-only samples T_l' available in the low-resource language.

2) Hunit: We begin by providing a brief overview of HuBERT to introduce Hunits. HuBERT is a self-supervised learning (SSL) approach that incorporates an offline clustering step to establish aligned target labels for a BERT-like prediction loss [29]. The backbone of HuBERT consists of a convolutional encoder, followed by multiple identical Transformer blocks [51] within a BERT mask predictor. HuBERT learns a combined acoustic and language model by applying the prediction loss exclusively to the masked regions within the continuous input sequence $X = [x_1, \ldots, x_T]$. The mask predictor, given the masked version $\hat{X}$, predicts a distribution over the target codewords at each time step t. Here, we denote C as the number of clustered codewords, $\mathbf{e}_c$ as the embedding for codeword c, and $\mathbf{A}_t$ as the output feature at step t. The distribution over codewords is formulated as follows:

$$p(c \mid \hat{X}, t) = \frac{\exp(\mathrm{sim}(\mathbf{A}_t \mathbf{W}, \mathbf{e}_c)/\tau)}{\sum_{c'=1}^{C} \exp(\mathrm{sim}(\mathbf{A}_t \mathbf{W}, \mathbf{e}_{c'})/\tau)} \tag{1}$$

In Equation (1), $\mathbf{W}$ represents a projection matrix, $\mathrm{sim}(\cdot, \cdot)$ computes the cosine similarity, and τ scales the logit.

Denote discrete target sequence for X as $Z = [z_1, z_2, \ldots, z_T]$, the prediction loss is formulated as:

$$\mathcal{L} = \sum_{t \in M} \log p \left(z_t \mid \hat{X}, t \right) \tag{2}$$

where $M \subset \{1 \ldots T\}$ are the masked timesteps for X.

The clustered codewords, which are iteratively refined and exhibit correlation with the underlying acoustic units, serve as universal phonetic units in our approach. We refer to these units as Hunits.

3) Hunit Extractor: As illustrated in Fig. 1(a), the Hunit extractor consists of three modules: a pretrained HuBERT model, a K-Means clustering module, and a denoising module. The denoising module incorporates a series of mode filters, commonly used in image segmentation tasks, to replace noisy elements within the clusters of HuBERT latent representations. By applying these filters, we reduce the noise in the Hunit sequence and produce reasonable articulatory boundaries.

4) Hunit Dictionary (HuDict): HuDict serves as a mapping from Hunit n-grams to speech fragments sourced from D_r. To establish frame-level alignment between each unpaired speech sample S_r in D_r and its corresponding Hunits H_r, we utilize the Hunit extractor. This alignment process, illustrated in Fig. 1(b), enables us to map Hunit n-grams to triplets *(utterance id, start frame, end frame)*, indicating the procedure for retrieving the corresponding speech fragment. These mappings are extracted from the frame-level alignment and added to HuDict. For practical implementation, we chose the n-gram mappings within the range $4 = n_{\min} \leq n \leq n_{\max} = 8$ based on several observations. Adhering to Theorem 1's constraint $n_{\min} \leq 2n_{\max}$ was primary. Additionally, a $n_{\min} \geq 5$ resulted in increased synthesis failures due to insufficient short speech segments. In contrast, a range of $n_{\min} = 3 \leq n \leq n_{\max} = 6$ led to reduced performance as shown in Table III, likely from overuse of short speech segments.

5) Grapheme-to-Hunit (G2H): Our G2H model leverages the lightweight SoundChoice [52] Grapheme-to-Phoneme (G2P) model built on the Conformer-Transformer architecture to train at the sentence level. As depicted in Fig. 1(c), paired speech data $\{S_l, T_l\} \in D_l$ allows us to generate training data $\{T_l, H_l\}$ for the G2H model by extracting the Hunit sequence H_l from S_l via the Hunit extractor. A key distinction is that H_l does not contain consecutive duplicates, unlike H_r from Fig. 1(b).

6) Audio Synthesis: The audio synthesis stage of our pipeline, illustrated in Fig. 1(d), begins by converting a text sample $T'_l \in D'_l$ into its corresponding Hunit sequence H'_l using the G2H model. Our objective is to decompose this sequence H'_l into an ordered collection of Hunit n-grams. To achieve this, we employ a divide-and-conquer strategy outlined in Algorithm 1. Such a strategy is designed to optimize the average length of the resulting Hunit n-grams, thereby facilitating the generation of synthesized speech with improved fluency. However, a direct implementation of the divide-and-conquer approach can lead to excessive redundant computations, which impede the efficiency of the algorithm, particularly for longer Hunit sequences.

To address this inefficiency, we introduce a caching system that utilizes a fixed-size priority queue. This cache stores the optimal decompositions of previously computed Hunit sequences, with priorities determined by the frequency of sequence occurrence. When the cache reaches its capacity, the least frequently occurring sequence is discarded to make space for new entries. This caching mechanism effectively prunes the divide-and-conquer procedure, accelerating the audio synthesis process while maintaining manageable memory consumption.

Mathematical induction shown in Theorem 1 guarantees that Algorithm 1 always returns Hunit n-gram sequences with the maximum averaged n if $n_{\max} \leq 2n_{\min}$, if such sequences exist. The $\oplus$ denotes the concatenation. Hunit sequences that cannot be decomposed into Hunit n-grams in HuDict are discarded during this process. The final step involves mapping the resulting H'_l n-gram sequences to speech fragments using HuDict and concatenating these fragments to generate the complete speech.

Theorem 1: Given an Hunit sequence $H = \{h_1, h_2, \ldots, h_m\}$ and a set of Hunit n-grams $\mathbb{S}$ with $n_{\min} \leq n \leq 2n_{\min}$, a greedy divide-and-conquer algorithm $\mathcal{D}$, which selects the first longest available n-gram from $\mathbb{S}$ present in H, guarantees a partition of H into a minimal number of Hunit n-grams, each of which belongs to $\mathbb{S}$.

Proof: Let $p(H)$ denote the minimal partition number of a sequence H. We prove by induction on m.

Base case: For an Hunit sequence H, where $|H| = n_{\min}$, the algorithm $\mathcal{D}$ will directly select H if $H \in \mathbb{S}$.

Induction Hypothesis: Assume that for all H, where $|H| < k$ for some $k > n_{min}$, $\mathcal{D}$ guarantees a partition of H into a minimal number of n-grams $\{s_1, s_2, \ldots, s_{p(H)}\}$, with $s_i \in \mathbb{S}$.

Now, consider an Hunit sequence H of length k.

Let $N \in \mathbb{S}$ be the longest n-gram appearing in H. The selection of N partitions H into three subsequences H_1, N, H_2, where $|H_1|, |H_2| < k$. The resulting n-grams are $\{s_1, s_2, \ldots, s_{p(H_1)+p(H_2)+1}\}$.

Assume another algorithm $\mathcal{D}'$ first selects M, where $M \notin H_1, M \notin H_2$. The original N is thus separated into at least two parts N_1, N_2. The number of resulting n-grams $P \geq p(H_1) - 1 + p(H_2) - 1 + 2 = p(H_1) + p(H_2)$. The inequality holds with equality if and only if $s' = s_{p(H_1)} \oplus N_1 \in \mathbb{S}$ and $s'' = N_2 \oplus s_{p(H_1)+2} \in \mathbb{S}$. However, this is not possible since either s' or s'' is longer than $|N|$ in this case for $n_{\min} \leq n \leq 2n_{\min}$, contradicting the assumption that N is the longest n-gram. Therefore, the number of resulting n-grams is at least $p(H_1) + p(H_2) + 1$, the same as the result produced by $\mathcal{D}$. ■

Algorithm 1: Decompose a Hunit Sequence Into Hunit N-Grams With Cache Pruning.

```
Input: $x$, the Hunit sequence
Input: $\mathbb{S}$, the set of all Hunit n-grams in the HuDict
Input: $n_{max}, n_{min}$ for Hunit n-grams
Input: $\mathbb{C}$, a mapping cache from Hunit sequences to
       their computed decompositions, initialized as
       empty
Output: $y$, list of Hunit n-gram sequences
function decompose($x, \mathbb{S}$)
    $y = []$
    if $\mathbb{C}$.contains($x$) then
        return $\mathbb{C}[x].ngrams$
    for $n \leftarrow n_{max}$ to $n_{min}$ do
        for $i \leftarrow 0$ to length($x$) $- n$ do
            if $x[i : i + n] \in \mathbb{S}$ then
                $y_{prev}$ := decompose($x[0 : i], \mathbb{S}$)
                if length($y_{prev}$) = 0 then
                    continue
                $y_{post}$ := decompose($x[i + n :], \mathbb{S}$)
                if length($y_{post}$) = 0 then
                    continue
                for $s_{prev} \in y_{prev}$ do
                    for $s_{post} \in y_{post}$ do
                        $s$ := $s_{prev} \oplus$ x[i:i+n] $\oplus s_{post}$
                        Append $s$ to $y$
        if length($y$) $\neq$ 0 then
            break
    if length($y$) $\neq$ 0 then
        $\mathbb{C}[x].ngrams := y$
    else
        $\mathbb{C}[x].ngrams := \varnothing$
    return $y$
```

C. Confidence Sampling for Improved Data Splicing

The fidelity of the selected speech segments to the target Hunits greatly impacts the quality of the synthesized speech in data splicing. To enhance this fidelity, we introduce a confidence sampling strategy that estimates the likelihood of speech segments given a Hunit n-gram and biases the selection towards segments with higher likelihood.

The likelihood of a Hunit n-gram c corresponding to a speech segment X' of a speech sequence X is computed as the average of the likelihoods of all the Hunits $\{c_1, c_2, \ldots, c_n\}$ in the n-gram. It can be formulated as:

$$L(c \mid X') = \frac{1}{n} \sum_{i=1}^{n} p(c_i \mid X, t + i) \tag{3}$$

Here, c_i represents the i-th Hunit in the n-gram, and $t + i$ is the timestamp corresponding to c_i in the original speech X of the speech segment X'.

To select a speech segment for a given Hunit n-gram, we calculate the likelihoods of all speech segments corresponding to that n-gram and apply a softmax function controlled by the temperature parameter τ' to create a probability distribution over these segments. The probability of selecting $X'^{(i)}$ among all speech segments $\mathbb{X}$ corresponding to Hunit-ngram c is thus:

$$f(X'^{(i)} \mid c) = \frac{\exp(L(c \mid X')/\tau')}{\sum_{X'^{(i)} \in \mathbb{X}} \exp(L(c \mid X'^{(i)})/\tau')} \tag{4}$$

Finally, we perform sampling from this distribution to select a speech segment, ensuring that segments with a higher correspondence to the Hunit n-gram are more likely to be chosen. The introduction of confidence sampling to the data splicing pipeline improves the quality of synthesized speech, thereby enhancing the performance of ASR and expediting model convergence.

TABLE I
Number of Words in D_l' for Different Languages

	Frisian	French	Dutch	German	Turkish
# words	275 K	5.5 M	563 K	6.6 M	289 K

III. Experiment Setup

A. Datasets

Our experimental setup utilizes a fine-tuned HuBERT model that was pretrained on the LibriSpeech dataset, using 10 hours of paired data from the CommonVoice dataset [53]. The CommonVoice dataset[1] is a multilingual speech corpus primarily sourced from Wikipedia articles. In this study, we focus on five languages: Frisian, French, Dutch, German, and Turkish.

As a reminder, we defined the notations D_r, D_l, and D_l' in Section II-B. Specifically, D_r represents unpaired speech data in the rich-resource language, D_l denotes paired low-resource data, and D_l' represents the set of available text transcriptions in the low-resource language.

For the cross-lingual data splicing experiments, we used the LibriSpeech dataset, which comprises 960 hours of speech, as our unpaired speech dataset D_r. Transcriptions were not utilized for this dataset. To emulate a low-resource scenario, we sampled a 10-hour subset denoted as D_l[2] from the CommonVoice dataset for each of the five chosen languages. Additionally, we used all available text transcriptions in the original training set of each language for D_l'. The number of words in each language in D_l' is provided in Table I. We use the official CommonVoice dev and test sets for performance testing.

To assess the efficacy of Hunit as a phonetic unit for data splicing, we conducted a monolingual data splicing experiment in English. We utilized the Libri-Light 10-hour (LL-10 h) setup, using the 10-hour paired data as D_r to build both the phoneme and Hunit dictionaries. The LL-10 h dataset was also used as D_l to train the G2H model. We experimented with different text sets for D_l', including the Libri-Light 10-hour text, LibriSpeech clean 100-hour text, and LibriSpeech 960-hour text.

To validate the scalability of our proposed method, we conducted another cross-lingual data splicing experiment using the

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

[2] utterance list released at https://github.com/IceCreamWW/SpliceTTS/tree/main/examples/assets/CommonVoice

TABLE II
DECODING HYPERPARAMETERS FOR DIFFERENT LANGUAGES

	Frisian	French	Dutch	German	Turkish
LM weight	7.0	4.0	3.2	4.6	3.7
word insert.	-0.1	-0.6	-0.8	-0.8	-0.4

larger Libri-Light medium dataset (LL-6 k) for constructing the Hunit dictionary, which contains 6,000 hours of unpaired speech data.

B. Hunit Extractor and G2H Model

The Hunit extractor in our framework is based on the officially released `HuBERT-base-iter2` checkpoint. [3] We obtained the latent representations from the ninth layer of the HuBERT model, which were then used to extract Hunits. These extracted Hunits were further clustered using a K-Means model trained on the latent representations from D_r.

For the Grapheme-to-Hunit (G2H) model, we adopted the Conformer-Transformer encoder-decoder architecture described in SoundChoice-G2P [52]. Instead of training the G2H model from scratch on D_l, we performed fine-tuning on a pretrained SoundChoice-G2P checkpoint. [4] Since the amount of data available in D_l is limited, fine-tuning the G2H model requires less than one hour on a single GPU. To ensure faster inference, we adopted a greedy decoding scheme.

C. HuBERT Model Fine-Tuning

We utilized the FAIRSEQ [54] toolkit to conduct model fine-tuning. The HuBERT model, consistent with the one used for Hunit extraction, comprised 12 transformer blocks with hidden dimensions of 768 and 8 attention heads. To enable end-to-end prediction of the output tokens, we introduced a randomly initialized output layer that was integrated with the pretrained encoder. We used the Connectionist Temporal Classification (CTC) loss as the optimization criterion. We followed the `base_10 h` fine-tuning setup described in HuBERT but extended the number of training steps from 10 k to 100 k. This adjustment was made to accommodate the longer convergence time observed in languages other than English during our experiments.

D. Decoding

We report results for Viterbi decoding and n-gram LM decoding. For n-gram LM decoding, we trained a 4-gram LM on D_l' for each of the five languages. The hyperparameters for decoding were tuned on the development set of each language. The decoding process employed a beam size of 500. The weights assigned to the LM and the word insertion penalty for decoding across different languages are provided in Table II.

TABLE III
RESULT (WER%) ON LIBRISPEECH TEST SETS OF DATA SPLICING WITHIN THE SAME LANGUAGE (VITERBI DECODING)

	Phonetic Units	Text	test clean	test other
1	N/A	N/A	9.7	16.7
2	Phonemes	LS-960	**8.2**	**15.8**
3		LS-clean-100	8.5	16.0
4		LL-10	9.4	16.5
5	KM-100	LS-960	9.5	16.5
6		LS-clean-100	9.9	16.6
7		LL-10	10.1	17.2
8	KM-200	LS-960	9.2	16.2
9		LS-clean-100	9.4	16.3
10		LL-10	9.8	17.0
11	KM-500	LS-960	8.4	16.1
12		LS-clean-100	8.7	16.2
13		LL-10	9.5	16.5
14	KM-500 (n_{min}=3,n_{max}=6)	LS-960	9.4	16.4
15		LS-clean-100	9.5	16.4
16		LL-10	9.7	17.0
17	KM-1000	LS-960	9.0	16.2
18		LS-clean-100	9.2	16.5
19		LL-10	9.7	16.8

IV. EXPERIMENT RESULTS

A. Self-Supervised Phonetic Units

In this section, we present the findings of our data splicing experiment conducted within the same language. The primary objective of this experiment was to validate the effectiveness of Hunits as practical phonetic units compared to phonemes, thus enabling their application in cross-lingual data splicing scenarios. Additionally, we investigated the impact of varying the number of clusters in the K-Means model (denoted as KM) during Hunit extraction, as well as the influence of the amount of text data used for the splicing procedure.

The results are summarized in Table III. The baseline was a pre-trained HuBERT model fine-tuned on the LL-10 h dataset without LM rescoring, as indicated in the first line. The experimental findings demonstrated an improvement in performance as the number of K-Means clusters during Hunit extraction increased from 100 to 500. With an increase in clusters, WER for the test set decreased, indicating an enhancement in the quality of the phonetic units. The increased number of clusters ensured that the phonetic units became more distinctive.

However, when the text data used for data splicing was limited to LL-10, the benefits were found to be modest in the case of KM-500, with degradation observed for KM-100 and KM-200. We observed that when the LL-10 h dataset was used for both the target text and the construction of the phonetic unit dictionary, the data splicing procedure tended to favor the selection of the original speech segments. This preference was due to the presence of long Hunit n-grams in the original speech segments that matched the corresponding text, thus limiting the potential gains from data splicing.

When comparing the performance of phonetic units, while the precision of Hunits under KM-500 was not as high as that

[3] [Online]. Available: https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt

[4] [Online]. Available: https://drive.google.com/drive/folders/13udm2iAoIlJVp6OqCK0OUZ-oSEN0PUtp

TABLE IV
DATA RATIO AND COMPARISON TO NEURAL TTS

	System	Decode	Mixing Ratio (real : synthetic)	Dev	Test
1	Baseline	Viterbi	N/A	15.3	15.0
2		4-gram LM		2.9	2.7
3	Speed Perturb	Viterbi	N / A	14.8	13.8
4		4-gram LM		2.6	2.5
5	VITS	Viterbi	1 : 1	17.1	16.7
6		4-gram LM		3.5	3.1
7	KM-500 Splice	Viterbi	1 : 2	13.1	12.7
8		Viterbi	1 : 1	12.5	12.3
9		Viterbi	1 : 1/2	12.2	11.8
10		Viterbi	1 : 1/3	12.5	11.9
11		4-gram LM	1 : 1/2	**2.0**	**2.0**

of phonemes, it displayed reasonably comparable performance, supporting its practical application. This confirmation was crucial for subsequent experiments involving cross-lingual data splicing using KM-500 units.

The observations from the experiment highlighted a significant performance decrement with smaller cluster sizes, such as those in the KM-100 and KM-200 configurations, especially when working with limited text data for data splicing. This phenomenon emphasized an inherent issue related to the quality of Hunits extracted when the number of clusters in K-Means is limited. In such scenarios, the generated Hunits lacked adequate distinctiveness, thereby inhibiting the formation of a reliable Hunit dictionary that could accurately map each Hunit to its corresponding speech segment.

Interestingly, the degree of performance improvement on the test clean set was consistently higher than that on the test other set across all conditions. This discrepancy can be attributed to the challenges in splicing noisy speech segments accurately compared to clean ones. The extraction of precise Hunits from noisy segments becomes more challenging, leading to generated speech-text pairs from noisy speech segments with an increased rate of transcription errors, thereby constraining the improvement seen on noisier test sets.

B. Data Ratio and Comparison to Neural TTS

In this experiment, we investigated the optimal balance between real and synthetic data for ASR training, focusing specifically on the Frisian language. The adjustment of the mixing ratio between real and synthetic data was achieved by repeating or subsetting D_l'. For example, a 1:1/2 mixing ratio meant subsetting D_l' to half in each epoch before mixing real and synthetic data into batches. The results in Table IV. indicated a preference for a 1:1/2 mixing ratio in the case of Frisian. Consequently, this ratio was adopted for the remaining languages in subsequent experiments.

We also compare the proposed method with speed perturb data augmentation in Table IV. We set the perturbation factors to 0.9, 1.0, and 1.1. The results, displayed in the third and fourth rows, indicate that speed perturbation yields a relative improvement of approximately 10% in WER.

Another noteworthy finding from this experiment was the negative impact on overall performance when the proportion of synthetic data was excessively increased as shown in the fifth and sixth line of Table IV. This degradation in performance could be attributed to the mismatch between the synthesized training data and the real-world test data. Despite the advanced methods used to create synthetic data, it failed to fully replicate the complex nature of real speech data, leading to a clear degradation in performance.

Furthermore, we included results for data synthesis using a VITS [55] neural TTS system trained on D_l with ESPnet [56] following the multi-speaker recipe[5], with the sample rate set to 16 kHz. The results revealed that the VITS model trained with 10 hours of noisy ASR training data produced low-quality audio that had a detrimental effect on ASR performance.

C. Data Splicing Across Different Languages

In this series of experiments, we conducted a comparative analysis across five languages to evaluate the effectiveness of data splicing using different numbers of KM clusters: KM-200, KM-500, and KM-1000. The results are presented in Table V.

The baseline model is a pre-trained HuBERT model fine-tuned on the 10-hour paired data for each of the five languages. A consistent trend observed across all five languages was the performance degradation when employing KM-200 for data splicing compared to the baseline results as shown in the third line to the sixth line. This highlights the insufficient quality of phonetic units produced by KM-200, indicating that these units are not universal enough for the proposed data splicing framework. On the other hand, KM-500 exhibited a substantial improvement in performance, demonstrating the effectiveness of our proposed method. Additionally, it also demonstrated synergistic compatibility with LM rescoring. The significant performance gain is preserved even after LM rescoring.

When the number of KM clusters was increased further to KM-1000, contrary to expectations, no additional gain was observed. In fact, KM-1000 performed less optimally than KM-500. We hypothesize that this can be attributed to the overly strict distinction of acoustic pieces by KM-1000. During data splicing, this heightened specificity makes it challenging to identify Hunit n-grams with a larger n, leading to the excessive concatenation of smaller speech segments as well as an increased amount of text that failed to be synthesized into speech. Consequently, this results in disfluency in the synthesized speech and less diversity in the synthesized dataset.

Notably, the performance improvement observed for Turkish was relatively lower compared to the other four languages. This discrepancy could potentially be attributed to the distinction in language families. While the other four languages belong to the Indo-European family, which aligns with the English language used for data splicing, Turkish represents a distinct language family. The contrasting language family categorization suggests that Turkish may encounter a greater degree of acoustic

[5] [Online]. Available: https://github.com/espnet/espnet/tree/master/egs2/TEMPLATE/tts1#vits-training

TABLE V
RESULTS (WER%) ON ALL FIVE LANGUAGES, τ' IS THE TEMPERATURE PARAMETER IN (4)

Phonetic Units		Confidence Sampling	Decode	Frisian		French		Dutch		German		Turkish	
				dev	test	dev	test	dev	test	dev	test	dev	test
1	N/A	N/A	Viterbi	15.3	15.0	42.8	46.9	20.9	22.1	36.5	39.1	28.1	29.2
2			4-gram LM	2.9	2.7	27.4	24.1	13.4	11.9	19.0	20.7	10.1	9.5
3	km-200	✗	Viterbi	16.8	16.4	45.9	50.6	22.7	24	39.6	41.8	30.1	31.8
4			4-gram LM	3.5	3.2	29.2	26.5	15.1	12.6	21.2	22.3	12.3	10.1
5		$\tau' = 0.2$	Viterbi	14.3	14.1	41.1	43.6	18.7	20.1	36.1	38.7	27.6	28.5
6			4-gram LM	2.4	2.2	24.5	23.9	12.7	13.5	16.7	17.7	9.4	9.2
7	km-500	✗	Viterbi	12.2	11.8	35.7	39.3	17.9	18.7	33.0	35.1	26.2	27.1
8			4-gram LM	2.0	2.0	21.1	23.6	10.7	9.2	14.7	15.8	8.6	7.9
9		$\tau' = 1.0$	Viterbi	12.0	11.7	34.7	38.3	17.6	18.1	32.1	33.9	25.4	26.3
10		$\tau' = 1.0$	4-gram LM	2.0	2.0	20.9	23.3	10.5	9.0	14.4	15.3	8.4	8.0
11		$\tau' = 0.2$	Viterbi	11.9	11.6	34.2	37.5	17.1	17.8	31.6	33.5	25.1	25.8
12		$\tau' = 0.2$	4-gram LM	**1.9**	1.9	**20.7**	**23.0**	**10.2**	8.9	14.2	**15.1**	8.4	7.8
13		$\tau' = 0.1$	Viterbi	11.8	11.6	34.5	37.9	17.2	17.6	31.3	33.2	24.8	25.2
14		$\tau' = 0.1$	4-gram LM	**1.9**	**1.8**	20.8	23.1	**10.2**	**8.8**	**14.0**	15.2	**8.2**	**7.5**
15	km-1000	✗	Viterbi	13.0	12.6	37.4	42.1	18.8	20.0	34.5	37.2	27.5	28.7
16			4-gram LM	2.3	2.1	23.0	25.7	11.3	10.5	15.6	16.7	9.4	8.7
17		$\tau' = 0.2$	Viterbi	12.8	12.4	37.0	41.8	18.5	19.6	34.1	36.6	27.3	28.4
18			4-gram LM	2.2	2.1	22.8	23.6	12.7	13.3	16.2	17.1	9.3	9.0

mismatch during the data splicing process, consequently placing limitations on the quality of the synthesized speech.

D. Confidence Sampling

In this series of experiments, we investigated the impact of confidence sampling on data splicing using different numbers of KM clusters (KM-200, KM-500, and KM-1000) across five languages. Additionally, for KM-500, we examined the influence of different values of τ' in (4) (1.0, 0.2, 0.1) used for confidence estimation.

A notable result was observed with confidence sampling applied to KM-200. Comparing the third and fourth line with the fifth and sixth line, confidence sampling effectively mitigated the initial performance degradation and even led to a slight improvement. This indicates that confidence sampling improved the discriminative power of KM-200 HUnits, addressing their inherent limitations. For KM-500, confidence sampling consistently improved performance across all test sets, as presented in the ninth to fourteenth lines of Table V. However, the benefits of confidence sampling were less pronounced for KM-1000. We speculate that the highly distinguishing nature of Hunits generated by KM-1000 contributed to this result. The challenges related to disfluency and over-reliance on small speech segments during concatenation cannot be addressed with confidence sampling.

The influence of the temperature parameter τ' on the confidence sampling strategy was investigated using three settings: 1.0, 0.2, and 0.1 as shown in the ninth to fourteenth lines of Table V. A high setting of 1.0 results in a smooth distribution, which does not completely filter out low-confidence segments. Despite this, a modest performance improvement over the baseline (without confidence sampling) is observed, indicating the potential benefits of even a moderate level of filtering. Decreasing the temperature to 0.2 significantly enhances performance, indicating effective filtration of low-quality samples. However, further decreasing τ' to 0.1 resulted in diverse outcomes. Although some instances showed additional performance improvements compared to $\tau' = 0.2$, these improvements lacked consistency compared to decreasing τ' from 1.0 to 0.2. This observation suggests that the segments retained after filtering at this temperature generally exhibit high quality and suitability for ASR training. However, the inconsistent gains indicate the existence of an optimal temperature setting beyond which the performance enhancements may diminish. In Fig. 3, we also illustrate the selection probability distribution under these τ' settings. As we decrease τ' from 1.0 to 0.1, the proportion of speech segments with notably low selection probability (specifically $p \in [0, 0.01)$) rises from 16.4% to 30.5%. This indicates an effective filtering out of suboptimal segments, consequently enhancing the quality of the synthesized speech.

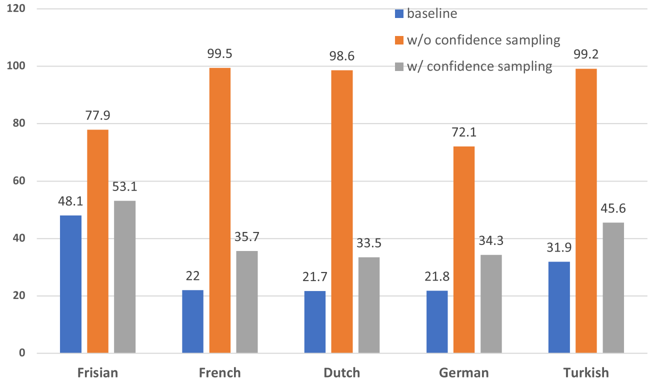

Fig. 2. Convergence Steps (K) with and without confidence sampling.

Another important observation was the efficiency gain achieved by incorporating confidence sampling. The data splicing process demonstrated faster convergence, as illustrated

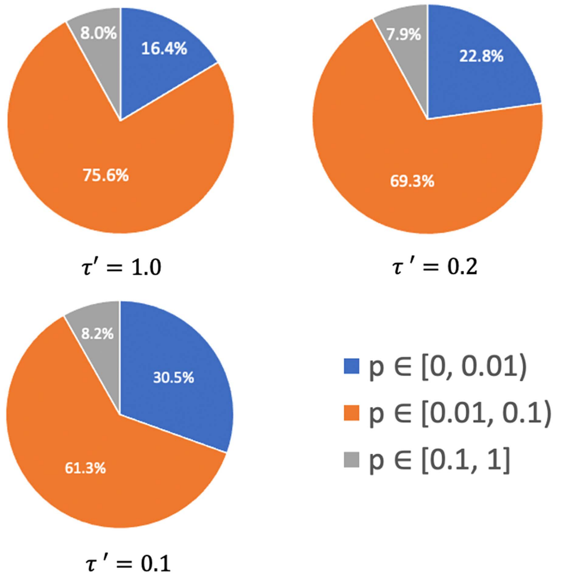

Fig. 3. Distribution of selection probabilities of speech segments with different τ'.

TABLE VI
RESULT (WER%) ON DIFFERENT SIZES OF SPEECH AND TEXT CORPORA

Language	D_r	D_l' (#words)	Decode	Dev	Test
Frisian	LS-960	50K	Viterbi	12.9	12.5
	LS-960	150K	Viterbi	12.1	11.9
	LS-960	275K	Viterbi	11.9	11.6
	LS-960	275K	4-gram LM	1.9	1.9
	LL-6K	275K	Viterbi	10.4	10.4
	LL-6K	275K	4-gram LM	1.6	1.7
French	LS-960	5.5M	Viterbi	34.2	37.5
	LS-960		4-gram LM	20.7	23.0
	LL-6K		Viterbi	31.0	33.1
	LL-6K		4-gram LM	18.3	21
Dutch	LS-960	563K	Viterbi	17.1	17.8
	LS-960		4-gram LM	10.2	8.9
	LL-6K		Viterbi	15.4	15.7
	LL-6K		4-gram LM	9.2	7.8
German	LS-960	6.6M	Viterbi	31.6	33.5
	LS-960		4-gram LM	14.2	15.1
	LL-6K		Viterbi	27.3	29.0
	LL-6K		4-gram LM	12.8	13.5
Turkish	LS-960	289K	Viterbi	25.1	25.8
	LS-960		4-gram LM	8.4	7.8
	LL-6K		Viterbi	22.3	23.3
	LL-6K		4-gram LM	7.5	7.0

Confidence sampling with τ'= 0.2 is applied to all experiments.

in Fig. 2. Note that since we run all the experiments for 10 k steps, the number of convergence steps requried is defined as when the validation loss reaches its minimum value. The baseline in Fig. 2 is the convergence steps required when finetuning only on real data (i.e. first row in Table. V). This improvement can be attributed to the filtering effect of confidence sampling, which helps eliminate low-confidence, suboptimal speech segments and ultimately enhances the quality of synthesized speech. Notably, the introduction of the confidence sampling strategy required only approximately 10 k additional steps for convergence, while significantly improving the ASR performance. These findings highlight the benefits of integrating confidence sampling into data splicing procedures.

E. Investigation on the Size of D_r and D_l'

This set of experiments aimed to explore the impact of different sizes of unpaired speech (D_r) and text corpora (D_l') on the data splicing process. Two setups of speech corpora were examined: the LibriSpeech 960-hour (LS-960) and the LibriLight medium 6k-hour (LL-6 K). Additionally, for the Frisian language, the effects of different text corpus sizes (50 K, 150 K, and 275 K) were investigated. The results are presented in Table VI. Notably, in the case of the Frisian language, increasing the size of the D_l' resulted in performance enhancements. However, the performance gains became less significant when D_l' grew from 150 K to 275 K, indicating the presence of a bottleneck likely caused by the limited size of D_r. To address this constraint and validate the scalability of the proposed data splicing framework, we expanded the unsupervised speech corpus D_r from the 960-hour to the 6k-hour setup. Subsequently, an examination across all five languages consistently showed about a further 10% relative improvement by incorporating LL-6 K as D_r.

V. CONCLUSION

In this study, we introduce a novel speech synthesis framework to address the enduring challenge of E2E ASR for low-resource languages. The cornerstone of this framework is self-supervised learning (SSL) units derived from a pretrained HuBERT model (Hunit), serving as universal phonetic units across different languages. These Hunits are based on an optimal setup determined through comprehensive exploratory experiments. Furthermore, our intra-lingual data splicing experiment using English confirmed the viability of Hunits as reasonable phonetic units, albeit with slightly less precision compared to phonemes. In contrast to traditional neural text-to-speech (TTS), which suffers from training with noisy ASR data and produces lower-quality speech segments, our framework provides a robust solution. The novelty of our framework also lies in embedding a confidence sampling strategy within the data splicing process. This enables the systematic exclusion of low-quality or imprecise speech segments, leading to a substantial enhancement in ASR model convergence and overall performance. Empirical results drawn from the COMMONVOICE dataset demonstrate notable improvements in ASR performance. The relative WER reduction range from 20% to 35% for multiple languages under a 10-hour low-resource setup. Furthermore, the scalability of our framework was validated by successfully incorporating a larger unsupervised speech corpus as the source of speech fragments, yielding an additional 10% relative improvement. Moreover, our proposed framework offers the important advantage of generating synthesized data in real-time during ASR training without any adverse impact on the training speed.

REFERENCES

[1] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, “Improved training of end-to-end attention models for speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 7–11.

[2] W. Chan, D. S. Park, C. Lee, Y. Zhang, Q. V. Le, and M. Norouzi, “SpeechStew: Simply mix all available speech recognition data to train one large neural network,” in *Workshop Mach. Learn. Speech Lang. Process.*, 2021.

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 28 492–28 518.

[4] J. Li, “Recent advances in end-to-end automatic speech recognition,” *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, 2022. [Online]. Available: http://dx.doi.org/10.1561/116.00000050

[5] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, “Transfer learning for speech recognition on a budget,” in *Proc. 2nd Workshop Representation Learn.,*2017, pp. 168–177.

[6] 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.

[7] B. Li et al., “Scaling end-to-end models for large-scale multilingual ASR,” in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 1011–1018.

[8] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, “Sequence-based multi-lingual low resource speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4909–4913.

[9] S. Tong, P. N. Garner, and H. Bourlard, “An investigation of deep neural networks for multilingual speech recognition training and adaptation,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 714–718.

[10] N. T. Vu, D. Imseng, D. Povey, P. Motlicek, T. Schultz, and H. Bourlard, “Multilingual deep neural network based acoustic modeling for rapid language adaptation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 7639–7643.

[11] Y. Qian and Z. Zhou, “Optimizing data usage for low-resource speech recognition,” *IEEE/ACM Trans. Audio, Speech Lang. Proc.*, vol. 30, pp. 394–403, 2022, doi: 10.1109/TASLP.2022.3140552.

[12] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12449–12460.

[13] W. -N. Hsu, B. Bolte, Y. -H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[14] S. Khorram, J. Kim, A. Tripathi, H. Lu, Q. Zhang, and H. Sak, “Contrastive siamese network for semi-supervised speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 7207–7211.

[15] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, “Applying wav2vec2.0 to speech recognition in various low-resource languages,” 2020, *arXiv:2012.12121*.

[16] A. Baevski, M. Auli, and A. R. Mohamed, “Effectiveness of self-supervised pre-training for speech recognition,” 2019, *arXiv:1911.03912*.

[17] J. Zhao and W.-Q. Zhang, “Improving automatic speech recognition performance for low-resource languages with self-supervised models,” *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1227–1241, Oct, 2022.

[18] S. Kim and M. L. Seltzer, “Towards language-universal end-to-end speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4914–4918.

[19] 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.

[20] C. Wang, J. Pino, and J. Gu, “Improving cross-lingual transfer learning for end-to-end speech recognition with speech translation,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4731–4735.

[21] 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.

[22] V. Joshi, R. Zhao, R. R. Mehta, K. Kumar, and J. Li, “Transfer learning approaches for streaming end-to-end speech recognition system,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2152–2156.

[23] Y. Qian and J. Liu, “MLP-HMM two-stage unsupervised training for low-resource languages on conversational telephone speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2013, pp. 1816–1820.

[24] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 1006–1010.

[25] D. S. Park et al., “Improved noisy student training for automatic speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2817–2821.

[26] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, “HuBERT: How much can a bad teacher benefit ASR pre-training?,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6533–6537.

[27] S. Chen et al., “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 1505–1518.

[28] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “Data2vec: A general framework for self-supervised learning in speech, vision and language,” in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 1298–1312.

[29] 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. Computat. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[30] 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.

[31] 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. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3765–3769.

[32] A. T. Liu, S.-W. Li, and H.-y. Lee, “TERA: Self-supervised learning of transformer encoder representation for speech,” *IEEE/ACM TASLP*, vol. 29, pp. 2351–2366, 2021.

[33] P. P. Parada, A. Dobrowolska, K. Saravanan, and M. Ozay, “pMCT: Patched multi-condition training for robust speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*2022, pp. 3779–3783.

[34] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, “Unsupervised cross-lingual representation learning for speech recognition,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2426–2430.

[35] A. Babu et al., “XLS-R: Self-supervised cross-lingual speech representation learning at scale,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 2278–2282.

[36] J. Li et al., “Developing RNN-T models surpassing high-performance hybrid models with customization capability,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, 2020, pp. 3590–3594.

[37] Y. Deng et al., “Improving RNN-T for domain scaling using semi-supervised training with neural TTS,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 751–755.

[38] A. Tjandra, S. Sakti, and S. Nakamura, “Listening while speaking: Speech chain by deep learning,” in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 301–308.

[39] F. Yue, Y. Deng, L. He, and T. Ko, “Exploring machine speech chain for domain adaptation and few-shot speaker adaptation,” 2021, *arXiv:2104.03815*.

[40] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, “Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end ASR systems,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5674–5678.

[41] S. Murthy, D. Sitaram, and S. Sitaram, “Effect of TTS generated audio on OOV detection and word error rate in ASR for low-resource languages,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 1026–1030.

[42] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. Moreno, and G. Wang, “TTS4pretrain 2.0: Advancing the use of text and speech in ASR pretraining with consistency and contrastive losses,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 7677–7681.

[43] W. Wang, Z. Zhou, Y. Lu, H. Wang, C. Du, and Y. Qian, “Towards data selection on TTS data for children’s speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6888–6892.

[44] A. Graves, “Sequence transduction with recurrent neural networks,” 2012, *arXiv:1211.3711*.

[45] 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.

[46] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Almost unsupervised text to speech and automatic speech recognition,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5410–5419.

[47] A. H. Liu, T. Tu, H. y. Lee, and L.-S. Lee, “Towards unsupervised speech recognition and synthesis with quantized speech representation learning,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7259–7263.

[48] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. J. Skerry-Ryan, “Semi-supervised training for improving data efficiency in end-to-end speech synthesis,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6940–6944.

[49] R. Zhao, J. Xue, J. Li, W. Wei, L. He, and Y. Gong, “On addressing practical challenges for RNN-Transducer,” in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 526–533.

[50] M. Dong, K. T. Lua, and H. Li, “A unit selection-based speech synthesis approach for mandarin chinese,” *J. Chin. Lang. Comput.*, vol. 16, pp. 135–144, 2006.

[51] A. Vaswani et al., “Attention is all you need,” in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[52] A. Ploujnikov and M. Ravanelli, “SoundChoice: Grapheme-to-phoneme models with semantic disambiguation,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 486–490.

[53] R. Ardila et al., “Common voice: A massively-multilingual speech corpus,” in *Proc. Int. Conf. Lang. Resour. Eval.*, 2020, pp. 4218–4222.

[54] M. Ott et al., “fairseq: A fast, extensible toolkit for sequence modeling,” in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 48–53.

[55] J. Kim, J. Kong, and J. Son, “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 5530–5540.

[56] S. Watanabe et al., “ESPnet: End-to-end speech processing toolkit,” in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2207–2211.

Wei Wang (Graduate Student Member, IEEE) received the B.S. degree from the Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China, in 2019. He is currently working toward the Ph.D. degree with Shanghai Jiao Tong University, Shanghai, China, under the supervision of Yanmin Qian. His current research mainly focuses on speech recognition.

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 a Full Professor. From 2015 to 2016, he was an Associate Research with the Speech Group, Cambridge University Engineering Department, Cambridge, U.K. He has authored or coauthored more than 200 papers in peer-reviewed journals and conferences on speech and language processing, including T-ASLP, Speech Communication, ICASSP, INTERSPEECH and ASRU. He has applied for more than 80 Chinese and American patents and won five championships of international challenges. His current research interests include automatic speech recognition and translation, speaker and language recognition, speech separation and enhancement, music generation and understanding, speech emotion perception, multimodal information processing, natural language understanding, deep learning and multi-media signal processing. He was the recipient of several top academic awards in China, including Chang Jiang Scholars Program of the Ministry of Education, Excellent Youth Fund of the National Natural Science Foundation of China, and the First Prize of Wu Wenjun Artificial Intelligence Science and Technology Award (First Completion). He was also the recipient of several awards from international research committee, including the Best Paper Award in Speech Communication and Best Paper Award from IEEE ASRU in 2019. He is also the Member of IEEE Signal Processing Society Speech and Language Technical Committee.