MAC: A unified framework boosting low resource automatic speech recognition

Anonymous authors
Paper under double-blind review

Abstract

We propose a unified framework for low resource automatic speech recognition tasks named meta audio concatenation (MAC). It is easy to implement and can be carried out in extremely low resource environments. Mathematically, we give a clear description of MAC framework from the perspective of bayesian sampling. We propose a broad notion of meta audio set for the concatenative synthesis text-to-speech system to meet the modeling needs of different languages and different scenes when using the system. By the proper meta audio set, we can integrate language pronunciation rules easily. Besides, this can also help reduce the difficulty of force alignment, improve the diversity of synthesized audio, and solve the OOV problem in synthesis. Extensive experiments have demonstrated the great effectiveness of MAC on low resource ASR tasks. For CTC greedy search, CTC prefix, attention, and attention rescoring decode mode in Cantonese ASR task, Taiwanese ASR task, and Japanese ASR task the MAC method can reduce the CER by more than 15%. Furthermore, in the ASR task, MAC beats wav2vec2 (with fine-tuning) on common voice datasets of Cantonese and gets really competitive results on common voice datasets of Taiwanese and Japanese. Among them, it is worth mentioning that we achieve a **10.9%** character error rate (CER) on the common voice Cantonese ASR task, bringing about **30%** relative improvement compared to the wav2vec2 (with fine-tuning).

1 Introduction

Automatic speech recognition (ASR) is a traditional task with wide application. Before deep learning became popular, HMM-GMM (Rodríguez et al., 1997) models, which have an elegant mathematical form, were widely used for ASR tasks. It performed well in some simpler speech recognition scenarios. And the most famous HMM-GMM speech recognition toolkit was Kaldi (Povey et al., 2011). As the rise of deep learning, a large number of end-to-end speech recognition models have emerged, such as Speech-transformer (Dong et al., 2018), Conformer (Gulati et al., 2020), Espnet (Watanabe et al., 2018), Wenet (Yao et al., 2021), LAS (Chan et al., 2015), etc. Furthermore, there are also many pretrained large models, such as vq-wav2vec (Baevski et al., 2019), wav2vec (Schneider et al., 2019) and wav2vec2 (Baevski et al., 2020) (Conneau et al., 2020).

However, whether for a HMM-GMM (Rodríguez et al., 1997) model or advanced end-to-end models such as wenet (Yao et al., 2021), obtaining a practical model of speech recognition often requires a large amount of data for the learning (hundreds of hours, if not tens of thousands of hours). While in many scenarios it is often difficult and expensive to get enough audio training data, such as dialects. A straightforward solution is to use TTS for data augmentation. There has been a lot of research on TTS data augmentation methods for ASR tasks, such as (Laptev et al., 2020; Rossenbach et al., 2020; Sun et al., 2020). In this work, we propose a unified framework for boosting low resource ASR tasks named MAC. The key in the MAC framework is leveraging a novel concatenative synthesis TTS system. It is worth mentioning that compared with former TTS for ASR tasks architectures, our MAC framework has the following advantages:

- MAC leverages a novel concatenative synthesis text-to-speech system to boost the ASR task, which can integrate language pronunciation rules as the prior knowledge. Furthermore, the process of generating audio has strong interpretability and is easy to be adjusted in the generation process;
- We propose a broad notion of meta audio set for the concatenative synthesis text-to-speech system, and meta audio set dimensions can be flexibly determined according to prior knowledge, model complexity budget, audio data size, etc. Using the proper meta audio set, the MAC framework has benefits such as reducing difficulty in force alignment, increasing diversity in synthesized audios, etc.
- There is no need to train additional TTS neural networks, and the generation process is just simple splicing, which is easy to implement and saves computing resources;
- Most importantly, MAC framework can be carried out in extremely low resource environments (e.g., training data less than 10h) without the help of additional labeled data. And MAC has a promising potential to model any low resource language as long as there is prior knowledge of the pronunciation rules of the language.

Lots of experiments have demonstrated the great effectiveness of our method in low resource ASR tasks. For the Cantonese ASR task, Taiwanese ASR task, and Japanese ASR task, the MAC framework can reduce the CER by more than 15%. Furthermore, MAC outperformed fine-tuning wav2vec2 on the Cantonese common speech dataset and obtained very competitive results on the Taiwanese and Japanese common speech datasets. Among them, we achieved a **10.9%** character error rate (CER) on the common spoken Cantonese ASR task, resulting in a significant relative improvement of about **30%** compared to fine-tuning wav2vec2 model.

Also, semi-supervised learning and transfer learning are currently two main ways to deal with the (labeled) data limit problem. However, both of the approaches will possibly face certain difficulties in some low resource scenarios. We will briefly discuss these below.

1.1 Semi-supervised learning

For semi-supervised learning, pseudo-label algorithms are widely used. There are many variants of pseudo label algorithms. For example, iterative pseudo label algorithm (Xu et al., 2020). But their basic ideas are using the pseudo labels to help to train the model.

Table 1 reported in Higuchi et al. (2022) demonstrated the effectiveness of iterative pseudo-label semi-supervised algorithms. The experiments are conducted on the LibriSpeech dataset Panayotov et al. (2015) and TED3 dataset Hernandez et al. (2018). LL-10h, LS-100h, LS-360h and LS-860h in Table 1 represent the different splits of the LibriSpeech dataset Panayotov et al. (2015) and the results are evaluated on dev-other split and test-other split of LibriSpeech dataset.

Table 1: Results of the iterative pseudo label based semi-supervised learning for ASR (reported in (Higuchi et al., 2022)). The experiments are conducted on the LibriSpeech dataset Panayotov et al. (2015) and TED3 dataset Hernandez et al. (2018) with a character error rate (CER) evaluation criteria.

Resource	Dev	Test
LS-100h	22.5	23.3
LS-100h/LS-360h	15.9	15.8
LS-100h/LS-860h	**13.9**	**14.2**
LL-10h	50.6	51.3
LL-10h/LS-360h	35.4	36.1
LL-10h/LS-860h	**33.5**	**34.4**

Besides the fact that the performance can be indeed improved greatly, we also notice that the quality of the initial model performance (the first row in Table 1) has a remarkable impact on the final performance. There

are also some theoretical results to explain these phenomena (see e.g., (Wei et al., 2020) and (Min & Tai, 2022)). Actually, in real low resource scenarios, it can be rather difficult to obtain a suitable initial model (pseudo-label generator) due to the limited labeled data. These difficulties seriously affect the performance of pseudo label based semi-supervised learning when applied to extremely low resource ASR tasks.

1.2 Transfer learning

For fine-tuning pretrained large models, wav2vec2 (Baevski et al., 2020; Conneau et al., 2020) is one of the most representative pretrained models. Wav2vec2 has demonstrated strong transfer learning capabilities on ASR tasks. Using wav2vec2, the CER can be significantly reduced (Yi et al., 2021). Although transfer learning is generally quite effective, the effectiveness of transfer learning can be significantly constrained, if there is a large gap between the target speech domain and the pretrained speech domain. For example, the wav2vec2 Baevski et al. (2020) model is pretrained on English audio data in the English speech domain, which gives a 4.8% CER by using 10 minutes of English audio labeled data to fine-tune. However, it only achieves a 28.32% CER even using 27k training utts in the Japanese domain (Yi et al., 2021).

1.3 Our contributions

The contributions can be summarized as follows:

- We propose MAC framework, which leverages a novel concatenative synthesis text-to-speech system to boost the ASR task. And it can integrate language pronunciation rules as prior knowledge.
- We innovatively proposed a broad notion of meta audio set, which helps MAC to integrate different prior knowledge and flexibly apply it to different scenarios.
- Mathematically, we give a clear description of MAC from the perspective of bayesian sampling.
- Our experiments show that MAC brings more than 15% reduction in CER in Cantonese ASR task, Taiwanese ASR task, and Japanese ASR task.

2 Relate work

There have been many attempts to use TTS for data augmentation to benefit ASR e.g. (Laptev et al., 2020; Rossenbach et al., 2020; Sun et al., 2020; Li et al., 2018; Ueno et al., 2021; Rosenberg et al., 2019; Tjandra et al., 2017). Among them, (Ueno et al., 2021) keep their eyes on the perspective of the representation. Results in Rosenberg et al. (2019) indicate the effectiveness of TTS data enhancement, although it may not be as good as the model trained on real speech data. (Tjandra et al., 2017) takes advantage of the close connection between TTS and ASR models.

The idea of leveraging concatenative synthesis text-to-speech to boost ASR also used in (Du et al., 2021; Zhao et al., 2021). However, in the audio splicing data augmentation method in (Du et al., 2021), they just replace the English audio part of the code-switching audio, which is a simple and preliminary splicing method. Therefore, the spliced audio diversity is limited, and it is difficult to introduce audio containing OOV (out of vocabulary) text in this spliced method. And Zhao et al. (2021) focus on adaptation to new domains and is out of scope of low resource tasks.

Recently, (Min et al., 2022) proposed a new interesting discovery. Experiments in Min et al. (2022) show that competitive performance on Mandarin ASR tasks can be achieved with only 10h of Mandarin audio data with the following steps:

- Train models on one Mandarin audio dataset and perform force alignment to get Mandarin character-audio pairs.
- Map the character to pinyin using character-pinyin dictionary to build pinyin-audio database.

- For transcription in another Mandarin audio dataset, query the audio clips corresponding to each Mandarin character and concatenate them.

- Perform energy normalization.

Although the Min et al. (2022) is focused on Mandarin audio ASR tasks, it does have many properties that are well-suited for low resource ASR tasks. For example, it does not require other labeled audio data to work, which is valuable in the low resource environment. Besides the synthesis process is efficient, interpretable, and convenient for human intervention.

Inspired by the (Min et al., 2022), we propose the MAC, leveraging a novel concatenative synthesis text-to-speech system to boost the ASR task performance. In (Min et al., 2022), the meta audio refers specifically to pinyin since only focuses on Mandarin ASR tasks. In this work, we propose a much more general concept of meta audio than (Min et al., 2022). By specific meta audio choice, we can utilize the prior knowledge of pronunciation rules in a large number of language scenes. These advantages make MAC ideal for extremely low resource language ASR tasks. Furthermore, the work in (Min et al., 2022), focusing on Mandarin, can be regarded as a special case under the MAC framework. Extensive experiments prove the great effectiveness of MAC, particularly on low resource ASR tasks.

3 Method

Our basic idea is to generate the audio data by concatenating the meta audios corresponding to given transcription texts and then train the ASR neural network using the generated audio data. In our method, we propose a much more general concept of meta audio set. Based on the meta audio set conception , our MAC can be expressed mathematically rigorously from the perspective of bayesian sampling.

3.1 General audio datasets

The mathematical formulation is as follows. Denote the audio wave space by $\mathcal{X}$, and the transcription text space by $\mathcal{Y}$, i.e., $\mathcal{X} = \{x : \text{all the audio waves}\}$, $\mathcal{Y} = \{y : \text{all the transcription texts}\}$.

In general, for ASR tasks, the labeled audio dataset consists of audio-transcription pairs $\{(x_i, y_i)\}_{i=1}^N$ sampled from a certain underlying distribution P, and $\{x_i\}_{i=1}^N \sim P_x$, $\{y_i\}_{i=1}^N \sim P_y$ with P_x, P_y as the marginal distribution of P.

Obtaining a practical speech recognition model often requires hundreds even thousands of hours of audio-transcription pairs for training. Unfortunately, getting audio-transcription pairs is usually expensive. And in many scenarios such as dialects, one can only access around ten hours of audio-transcription pairs. However, the audio-only data $x \sim P_x$ and text-only data $y \sim P_y$ are respectively easier to access. Generally, we have a paired dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, an audio-only dataset $\mathcal{D}_{\text{audio}} = \{x_i\}_{i=1}^{N_1}$ and a text-only dataset $\mathcal{D}_{\text{text}} = \{y_i\}_{i=1}^{N_2}$ with $N_2 > N_1 > N$. This is the typical setting of low recourse ASR tasks. Now our goal is to sample paired audio-transcription data (x, y) from the underlying distribution P.

Basically, we have

$$P(x, y) = P_y(y)P(x \,|\, y), \tag{1}$$

where $P_y(y)$ corresponds to the distribution of transcriptions, and $P(x \,|\, y)$ denotes the distribution of audios given a certain transcription y. Therefore, the sampling of new data (audio-transcription pair) (x, y) can be divided into two stages. First, the transcription text y is sampled from $P_y(y)$, and then the audio x corresponding to the previous transcription text y is sampled from $P(x \,|\, y)$. The first stage, i.e., sampling of transcription text y, is relatively easy since usually there is sufficient text-only data in $\mathcal{D}_{\text{text}} = \{y_i\}_{i=1}^{N_2}$, and we can model/approximate P_y by $\tilde{P}_y(y) = \frac{1}{N_2}\sum_{y_i \in \mathcal{D}_{\text{text}}} \delta(y_i)$ (i.e., substitute the probability by sample frequency). Therefore, the key is to analyze and maximize $P(x \,|\, y)$, which is in fact a TTS task.

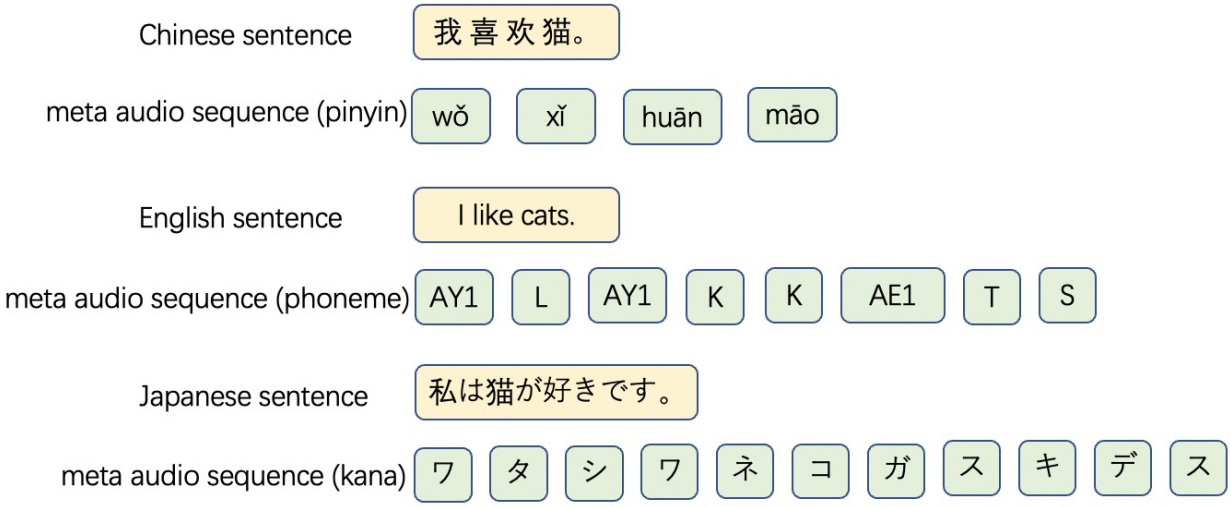

Figure 1: Examples of the mapping $\mathbf{t}$ in English, Japanese and Chinese. The function $\mathbf{t} : \mathcal{Y} \to \mathcal{A}$ is to map a transcription text to its corresponding meta audio sequence which reflects language-specific pronunciation rules. Hence the $\mathbf{t}$ can help to fusion prior knowledge of pronunciation rules. Obviously, different languages may have different mappings $\mathbf{t}$.

3.2 Meta audio set and meta audio sequence space

In order to perform further "decoupling" analysis on the conditional probability distribution $P(x \,|\, y)$, we first introduce the meta audio set. Here the meta audios refer to the basic modeling units of specific language pronunciations. For example, there are about 50 phonemes in English. If we want to use these phonemes as modeling units to characterize English pronunciation, these phonemes form a nature meta audio set of English. In this way, it is similar to monophonic modeling in HMM-GMM Rodríguez et al. (1997) and Kaldi Povey et al. (2011). However, in our framework, the meta audio set selections are flexible. For instance, a fusing phoneme (treating several phonemes as the same phoneme) set can be seen as a meta audio set in English. For a specific language, we can reasonably determine the meta audio according to its pronunciation rules e.g. pinyin in Mandarin and kana in Japanese.

The meta audio sequence space $\mathcal{A}$ represents sequences set of meta audios. Certainly, different meta audio sets may imply different meta audio sequence space $\mathcal{A}$.

Mapping function $\mathbf{t} : \mathcal{Y} \to \mathcal{A}$. To reflect language-specific pronunciation rules, we also need a function $\mathbf{t} : \mathcal{Y} \to \mathcal{A}$ to map a transcription text to its corresponding meta audio sequence. The construction of $\mathbf{t}$ requires prior knowledge of pronunciation rules. Obviously, different languages may have different mappings $\mathbf{t}$ and even different meta audio sequences $\mathcal{A}$ in the same language may have different mappings $\mathbf{t}$. Figure 1 shows examples of $\mathbf{t}$ for English and Japanese.

3.3 Decoupling analysis

In this section, we perform a fine-grained decoupling analysis on the probability $P(x \,|\, y)$, the conditional distribution of audios given transcriptions. On the one hand, when an audio $x \in \mathcal{X}$ is given, we have a conditional distribution $P(y \,|\, x)$. This is in fact the goal of ASR tasks: predict corresponding texts given audios by estimating $P(y \,|\, x)$. On the other hand, when a transcription $y \in \mathcal{Y}$ is given, we have another conditional distribution $P(x \,|\, y)$. We can decouple $P(x \,|\, y)$ via the meta audio sequence space $\mathcal{A}$ and the

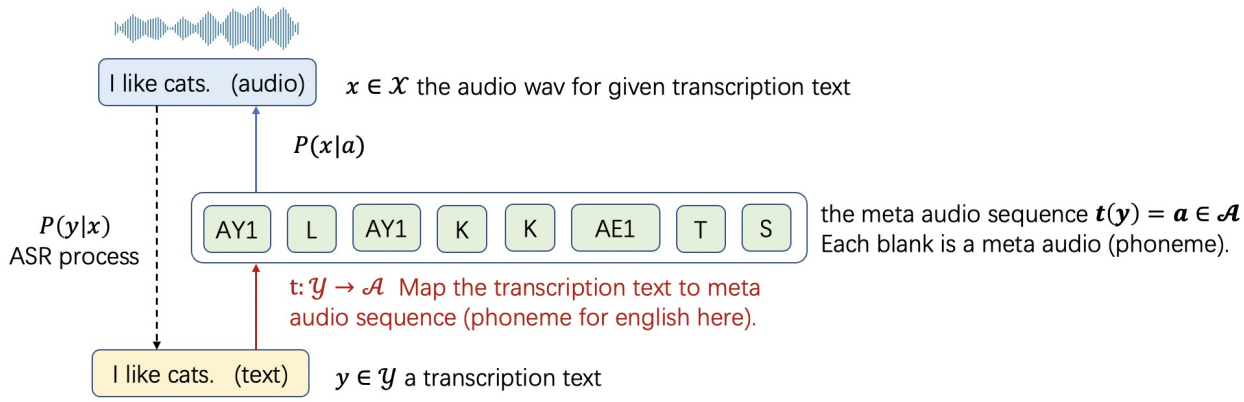

Figure 2: An illustration of the decoupling process. The goal of ASR task is to estimate $P(y\,|\,x)$ and the goal of TTS task is to estimate $P(x\,|\,y)$. In order to simplify the modeling of $P(x\,|\,y)$, we introduce a mapping $\mathbf{t}$ to transform $P(x\,|\,y)$ into $P(x\,|\,a)$. The dimension of a is generally much smaller than y.

mapping function $\mathbf{t}$:

$$\begin{aligned} P(x\,|\,y) &= \sum_{a\in\mathcal{A}} P(x, a\,|\,y) \\ &= \sum_{a\in\mathcal{A}} P(x\,|\,a, y)P(a\,|\,y) \\ &= P(x\,|\,a = \mathbf{t}(y), y) \end{aligned} \tag{2}$$

Here, we applied the fact that $P(a\,|\,y)$ is a degenerate distribution with the probability 1 at $a = \mathbf{t}(y)$. Furthermore, the meta audio sequence contains all the pronunciation information of the transcription text, hence

$$P(x\,|\,a = \mathbf{t}(y), y) = P(x\,|\,a = \mathbf{t}(y)), \tag{3}$$

which gives

$$P(x\,|\,y) = P(x\,|\,a = \mathbf{t}(y)). \tag{4}$$

Figure 2 illustrates this decoupling analysis.

3.4 Bayesian inference on $P(x\,|\,a)$

Based on Section 3.3 and Eq. (4), instead of analyzing $P(x\,|\,y)$ directly, we can turn to study $P(x\,|\,a)$. In general, this is much easier, since $a \in \mathcal{A}$ usually has a much lower dimension than $y \in \mathcal{Y}$. The dimension here refers to the number of element classes per position of $a \in \mathcal{A}$ or $y \in \mathcal{Y}$. For example, in English, the dimension of $a \in \mathcal{A}$ is about 50 (here we naturally select the phoneme as meta audio for simple interpretation), while the dimension of $y \in \mathcal{Y}$ may be much higher since there is a large number of words.

Notice that

$$P(x\,|\,a) \propto P_x(x)P(a\,|\,x), \tag{5}$$

The goal is now converted to maximize $P_x(x)$ and $P(a\,|\,x)$ in order to maximize $P(x\,|\,a)$. Recall that $P_x(x)$ reflects the prior probability of audios $x \in \mathcal{X}$, we will discuss it later (in Section 3.6). For $P(a = (a^{(1)}, a^{(2)}, ..., a^{(n)})\,|\,x)$, we have

$$P(a\,|\,x) = \sum_{\mathbf{s}} \prod_{i=1}^{n} P\left(a^{(i)}\,|\,x^{(i)} = [x_{s_i}, x_{s_{i+1}})\right) \tag{6}$$

Here, $\mathbf{s} = (s_1, s_2, ..., s_{n+1})$ represents the time slice of $x \in \mathcal{X}$. Eq. (6) holds because: 1) the audio wave x can be also properly divided (maybe not unique) to obtain the clip $x^{(i)} = [x_{s_i}, x_{s_{i+1}})$ corresponding to

each $a^{(i)}$. For instance, we can segment an English audio wave of one sentence and get the audio wave segmentation corresponding to the sentence's meta audio sequence (here meta audio sequence is phoneme sequence since here we naturally select the phoneme as meta audio for simple interpretation); 2) the audio wave clip $x^{(i)}$ in x is monotonous with respect to $a^{(i)}$ in a. That is, the timestamps of audio waves and meta audios must match with each other, i.e., $x^{(i)}$ and only $x^{(i)}$ corresponds to $a^{(i)}$; 3) For simplicity, we treat this correspondence as independent.

Unfortunately, it is quite expensive to consider all the time slices $\mathbf{s} = (s_1, s_2, ..., s_{n+1})$ in Eq. (6). However, for each fixed time slice $\mathbf{s}^0 = (s_1^0, s_2^0, ..., s_{n+1}^0)$, we can obtain a lower bound of $P(a \,|\, x)$:

$$\begin{aligned} P(a \,|\, x) &= \sum_{\mathbf{s}} \prod_{i=1}^{n} P\left(a^{(i)} \,|\, x^{(i)} = [x_{s_i}, x_{s_{i+1}})\right) \\ &\geq \prod_{i=1}^{n} P\left(a^{(i)} \,|\, x^{(i)} = [x_{s_i^0}, x_{s_{i+1}^0})\right). \end{aligned} \tag{7}$$

3.5 Maximization of $P(a \,|\, x)$

According to Eq. (7) in Section 3.4, we can approximately maximize $P(a = (a^{(1)}, a^{(2)}, ..., a^{(n)}) \,|\, x)$ by maximizing a lower bound determined by some fixed partition $\mathbf{s}^0 = (s_1^0, s_2^0, ..., s_{n+1}^0)$. The maximization of right hand side of Eq. (7) can be considered in another direction: for each element $a^{(i)}$ in a, we are required to find an audio wave clip $x^{(i)}$ that makes $P\left(a^{(i)} \,|\, x^{(i)}\right)$ as large as possible. Fortunately, this can be achieved by performing force alignment (Kim et al., 2021; López & Luque, 2022). Specifically, we first map the transcription text $y \in \mathcal{Y}$ in the labeled audio dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ into $a \in \mathcal{A}$ to get a corresponding dataset $\{(x_i, a_i)\}_{i=1}^N$. Then, we train the ASR model and perform force alignment on $\{(x_i, a_i)\}_{i=1}^N$ to get the audio wave clip corresponding to the meta audio element for each a_i (with high probability). For further efficiency, we can store the forced alignment results and build a database $\mathcal{B}$. Therefore, when we aim to maximize $\prod_{i=1}^n P\left(a^{(i)} \,|\, x^{(i)} = [x_{s_i^0}, x_{s_{i+1}^0})\right)$ for some fixed time slice $\mathbf{s}$, we just query the audio clip $x^{(i)}$ corresponding to each meta audio element $a^{(i)}$. Here, the time slice $\mathbf{s}$ is considered implicitly, since when we concatenate these audio clips, it automatically forms a complete time of the audio wave. The procedure of building the database $\mathcal{B}$ is illustrated in Figure 3.

Remark 1 ***(Database size)*** *For each element $a^{(i)}$, we may get different corresponding audio clips (with high probability) by conducting force alignment on $\{(x_i, a_i)\}_{i=1}^N$. We just store all of them in the database $\mathcal{B}$ to enrich the selections and increase the diversity of synthesized audios.*

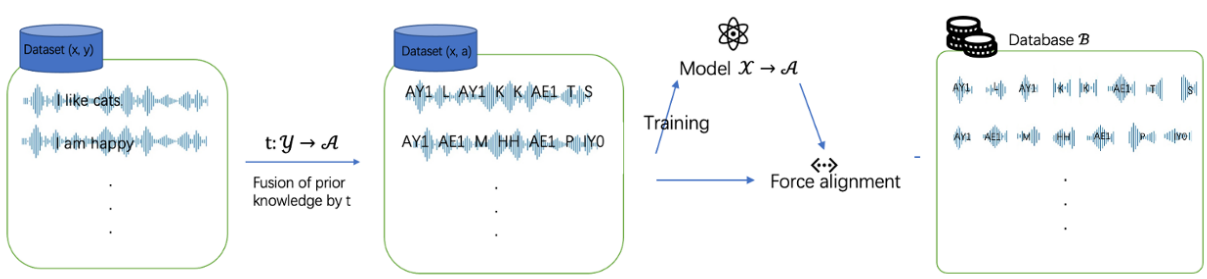

Figure 3: The process of building the database $\mathcal{B}$. Note that we can get audio clips from the training set, so we can train an ASR model on the training set and use that model for the forced alignment task on the (same) training set, which is very helpful in extremely low-resource conditions to obtain high-quality audio clips.

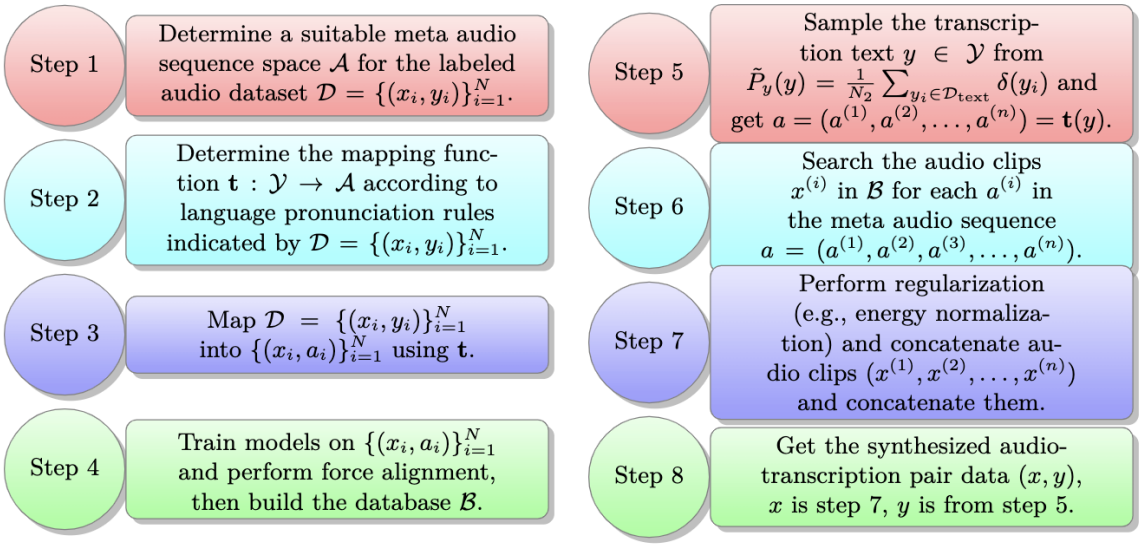

Figure 4: The MAC framework steps

3.6 Maximization of $P_x(x)$

In this section, we discuss about maximizing $P_x(x)$. In Section 3.5, for each element $a^{(i)}$ in a, we need to find an audio clip $x^{(i)}$ to enlarge $P\left(a^{(i)} \,|\, x^{(i)}\right)$ as much as possible. However, the audio wave obtained by combining these audio clips may give a higher probability $\prod_{i=1}^n P\left(a^{(i)} \,|\, x^{(i)}\right)$, but negatively affect $P_x(x = (x^{(1)}, x^{(2)}, ..., x^{(n)}))$. For example, the volume of generated $(x^{(1)}, x^{(2)}, ..., x^{(n)})$ may change rapidly and frequently, leading to a low $P_x(x = (x^{(1)}, x^{(2)}, ..., x^{(n)}))$. The reason can be mathematically understood that the support set of P_x is likely to be a very small (proper) subset in the whole audio space $\mathcal{X}$. Hence, simply merging these audio clips may cause the synthesized audio wave $x = (x^{(1)}, x^{(2)}, ..., x^{(n)})$ corresponding to $a = (a^{(1)}, a^{(2)}, ..., a^{(n)})$ to be severely distorted.

To solve these problems, we need some regularization techniques to improve the modeling of $P_x(x = (x^{(1)}, x^{(2)}, ..., x^{(n)}))$. A simple but effective regularization is the energy normalization (Li et al., 2002; Ahadi et al., 2004; Min et al., 2022), and here we just imitate the operation as in (Min et al., 2022). That is, averaging the energy of sampled audio clips $(x^{(1)}, x^{(2)}, ..., x^{(n)})$ corresponding to the meta audio sequence $a = (a^{(1)}, a^{(2)}, ..., a^{(n)}) \in \mathcal{A}$. Mathematically, we have

$$E = \frac{\sum_{i=1}^n \|x^{(i)}\|}{n}, \tag{8}$$

$$\left\{x^{(i)}\right\}_{i=1}^n \to \left\{\frac{x^{(i)}}{\|x^{(i)}\|} * E\right\}_{i=1}^n. \tag{9}$$

Since the energy normalization in Eq. (9) only linear scaling $x^{(i)}, i = 1, 2, ..., n$, we further have

$$P\left(a^{(i)} \,|\, x^{(i)}\right) \approx P\left(a^{(i)} \,|\, \frac{x^{(i)}}{\|x^{(i)}\|} * E\right). \tag{10}$$

Certainly, other (more complex) regularization operations are also widely used in concatenative synthesis-based text-to-speech (TTS) systems (Tabet & Boughazi, 2011; Khan & Chitode, 2016). In our experiments, the energy normalization is easy to implement and works well. More importantly, our ultimate goal is to do ASR tasks, so we do not pay too much attention to the fine-grained quality of the audio generated by

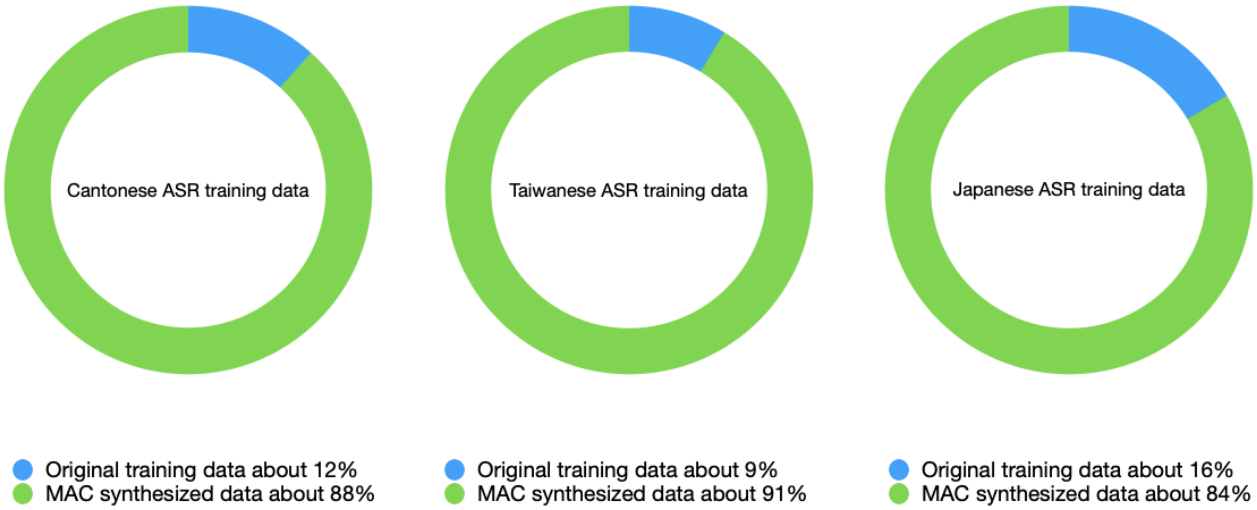

Figure 5: In the Cantonese ASR, Taiwanese ASR, and Japanese ASR three experimental tasks, the proportion of the audio data synthesized by MAC and the original audio data when we use MAC method.

TTS which may lead to overly complex models. However, improving the quality of the synthesized audio may give better results and we will further discuss the applicability of other (more complex) regularization operations in the future work.

3.7 Workflow of MAC

According to previous sections, the MAC framework steps can be summarized as Fig 4.

4 Experiments

We verify the effectiveness of MAC on three real low resource Cantonese, Taiwanese, and Japanese ASR tasks on the corresponding dataset of them in common voice dataset [1]. The experimental results show that MAC has defeated the results achieved by fine-tuning large-scale wav2vec2 pretrained models, and achieved very competitive results on these tasks. More importantly, MAC brings a remarkable performance boost on some tasks. For instance, we have achieved **10.9% CER** on the common voice dataset of Cantonese for ASR tasks, leading an around **30% relative improvement** compared to fine-tuning wav2vec2 model.

4.1 Datasets

Common voice is currently one of the most widely used multilingual audio datasets, which is a public speech dataset contributed by volunteers around the world. For each language, we use the training split of the corresponding dataset in common voice dataset as the origin labeled data to conduct MAC. The proportion of origin labeled data and synthetic data by MAC are shown in Fig. 5. For comparison, we also train directly on origin train split labeled data as the baseline results. We use the test split of the corresponding dataset in common voice dataset to evaluate.

[1] We use the zh-HK dataset in the common voice dataset for the Cantonese ASR task.

Table 2: CER for Cantonese, Taiwanese and Japanese on ASR tasks. We use the advanced hybrid CTC/attention model as the baseline (tested for four decoding modes: CTC greedy search, CTC prefix beam search, attention, and attention rescoring). It is shown that MAC can boost performance significantly, and also achieve competitive results compared to fine-tuning the pretrained wav2vec2 model.

task	model	decode mode	CER
Cantonese ASR	Hybrid CTC/attention model	CTC greedy search	32.5
		CTC prefix beam search	32.5
		attention	44.1
		attention rescoring	30.6
	Hybrid CTC/attention model+MAC	CTC greedy search	12.7
		CTC prefix beam search	12.7
		attention	11.0
		attention rescoring	**10.9**
	wav2vec2+fine-tuning	-	15.4
Taiwanese ASR	Hybrid CTC/attention model	CTC greedy search	51.3
		CTC prefix beam search	51.3
		attention	55.3
		attention rescoring	48.2
	Hybrid CTC/attention model+MAC	CTC greedy search	22.0
		CTC prefix beam search	22.0
		attention	18.6
		attention rescoring	19.5
	wav2vec2+fine-tuning	-	**18.4**
Japanese ASR	Hybrid CTC/attention model	CTC greedy search	45.3
		CTC prefix beam search	45.3
		attention	72.1
		attention rescoring	44.5
	Hybrid CTC/attention model+MAC	CTC greedy search	25.1
		CTC prefix beam search	25.0
		attention	24.3
		attention rescoring	**23.4**
	wav2vec2+fine-tuning	-	24.9*

4.2 Models

We use the advanced hybrid CTC/Attention architecture Watanabe et al. (2017) with the conformer Gulati et al. (2020) encoder as Wenet [2] (Yao et al., 2021). It is worth mentioning that in the evaluation phase, the final output is jointly determined by both CTC-decoder and attention-decoder for better performance when we use the attention rescoring decode mode. That is, the attention decoder rescores the top candidates given by the CTC-decoder. We conduct experiments on 2 $\times$ RTX 3090GPUs (24GB) and 4 $\times$ P100GPUs (16GB).

4.3 Details and results

Below we present the experimental details and results

We can see that our MAC method substantially exceeds the fine-tuning wav2vec2 model result on the Cantonese recognition task and is comparable to the wav2vec2 model results on other tasks.

[2]We refer to some codes from `https://github.com/wenet-e2e/wenet`

4.3.1 Cantonese ASR

For Cantonese, the natural choice of the meta-audio set is Cantonese pinyin at first glance. But taking advantage of the broad notion of the meta audio set, we make some adjustments on the Cantonese pinyin to simplify our modeling process, including but not limited to not distinguishing between different tones in Cantonese pinyin in the meta audio set. e.g. regarding "haa4" and "haa6" as one meta audio in the meta audio set. The function $\mathbf{t}$ is to map a Cantonese transcription text y to the meta audio sequence $a \in \mathcal{A}$. For a Cantonese transcription text $y = (y^1, y^2, ..., y^m)$, we generally have the approximation

$$\mathbf{t}(y^1, y^2, ..., y^m) \approx (\mathbf{t}(y^1), \mathbf{t}(y^2), ..., \mathbf{t}(y^m)). \tag{11}$$

Hence, we can reverse the order of Step 3 and Step 4 in Section 3.7 to simplify the overall process. That is, we first train the model on original labeled dataset (x, y) and perform forced alignment. And then pass the results through the mapping function $\mathbf{t}$ to build the database $\mathcal{B}$ for Cantonese. In this way, we can use the trained baseline Hybrid CTC/attention model to do force alignment and do not need to train another model for force alignment. We use transcriptions in the validation split of the zh-hK dataset in common voice dataset and remove transcriptions appearing in the test split as the text-only dataset $\mathcal{D}_{\text{text}}$ for Step 5. For Step 7, we use the energy normalization method described in Eq. (8) and (9).

4.3.2 Taiwanese ASR

For Taiwanese, the natural choice of the meta-audio set is pinyin at first glance. But taking advantage of the broad notion of the meta audio set, we make some adjustments on the Cantonese pinyin to simplify our modeling process, including but not limited to simplification of some polyphonic characters. The function $\mathbf{t}$ is to map a Taiwanese transcription text $y \in \mathcal{Y}$ to a meta audio sequence $a \in \mathcal{A}$. The pinyin mapping function $\mathbf{t}$ also has approximation Eq. (11) and hence we can also reverse the order of Step 3 and Step 4 to simplify the overall process. That is as Cantonese task above, we first train the model on original labeled dataset (x, y) and perform forced alignment, and then pass the results through the mapping function $\mathbf{t}$ to build the database $\mathcal{B}$ for Taiwanese. In this way, we can use the trained baseline Hybrid CTC/attention model to do force alignment and do not need to train another model for force alignment. We use transcriptions in the validation split of the taiwan dataset in common voice dataset and remove transcriptions appearing in the test split as the text-only dataset $\mathcal{D}_{\text{text}}$ for Step 5. For Step 7, we use the energy normalization method described in Eq. (8) and (9).

4.3.3 Japanese ASR

For Japanese, the natural choice of the meta-audio set is kana at first glance. But taking advantage of the broad notion of the meta audio set, we make some adjustments on the Cantonese pinyin to simplify our modeling process, including but not limited to simplification of some special kanas. The function $\mathbf{t}$ is to map a Japanese transcription text $y \in \mathcal{Y}$ to a kana sequence $a \in \mathcal{A}$. Here, we do not reverse Step 3 and Step 4. We use transcriptions in the validation split of the japan dataset in common voice dataset and remove transcriptions appearing in the test split as the text-only dataset $\mathcal{D}_{\text{text}}$ for Step 5. For Step 7, we use the energy normalization method described in Eq. (8) and (9).

4.3.4 Performance

All results are shown in Table 2. We just refer to fine-tuning wav2vec2 results from urls [3] so we did not apply the decode modes (CTC greedy search, CTC prefix beam search, attention, attention rescoring) to them. Firstly, one can observe a significant performance improvement by MAC compared to the advanced hybrid CTC/attention baseline model. For **all** CTC greedy search, CTC prefix, attention, and attention rescoring decode mode in **all** three Cantonese ASR task, Taiwanese ASR task, and Japanese ASR task, the MAC method can reduce the CER by more than 15%.

[3] we refer the fine-tuning wav2vec2 results on urls. Cantonese: `https://huggingface.co/ctl/wav2vec2-large-xlsr-cantonese` and Japanese: `https://huggingface.co/qqhann/w2v_hf_jsut_xlsr53` and Taiwanese: `https://huggingface.co/voidful/wav2vec2-large-xlsr-53-tw-gpt` and we round to one decimal place.

More importantly, MAC beats wav2vec2 (with fine-tuning) and achieves new SOTA on common voice datasets of Cantonese and Japanese when applied to ASR tasks. * in Table 2 represents that the 24.9 CER result is achieved by using *extra data* than the Japanese audio dataset in the common voice dataset during finetuning wav2wev2. We can see that MAC relatively improves the performance by about 30% on the Cantonese ASR task and even beats wav2vec2 fine-tuned with extra data on the Japanese ASR task. In addition, for the Taiwanese ASR task, MAC also achieves results comparable to the fine-tuned wav2vec2 model.

5 Conclusion

MAC proposed in this work is a unified framework for low resource automatic speech recognition tasks. We propose a broad notion of meta audio set to help MAC to be available as long as we have the knowledge of pronunciation rules to construct a suitable meta audio set. Besides, we give a clear mathematical description of MAC framework from the perspective of bayesian sampling. Our experiments have demonstrated the great effectiveness of MAC on low resource speech recognition tasks, with remarkable improvements in accuracy even without tuning hyperparameters carefully. We hope that the MAC method can contribute to the development of low resource audio recognition tasks.

References

Mohammad Ahadi, Hamid Sheikhzadeh, Robert Brennan, and George Freeman. An energy normalization scheme for improved robustness in speech recognition. In *Eighth International Conference on Spoken Language Processing*, 2004.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*, 2019.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*, 2020.

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888. IEEE, 2018.

Chenpeng Du, Hao Li, Yizhou Lu, Lan Wang, and Yanmin Qian. Data augmentation for end-to-end code-switching speech recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 194–200. IEEE, 2021.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve. Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *International conference on speech and computer*, pp. 198–208. Springer, 2018.

Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, and Takaaki Hori. Momentum pseudo-labeling: Semi-supervised asr with continuously improving pseudo-labels. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1424–1438, 2022.

Rubeena A Khan and Janardan Shrawan Chitode. Concatenative speech synthesis: A review. *International Journal of Computer Applications*, 136(3):1–6, 2016.

Jaeyoung Kim, Han Lu, Anshuman Tripathi, Qian Zhang, and Hasim Sak. Reducing streaming asr model delay with self alignment. *arXiv preprint arXiv:2105.05005*, 2021.

Aleksandr Laptev, Roman Korostik, Aleksey Svischev, Andrei Andrusenko, Ivan Medennikov, and Sergey Rybin. You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation. In *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 439–444. IEEE, 2020.

Jason Li, Ravi Gadde, Boris Ginsburg, and Vitaly Lavrukhin. Training neural speech recognition systems with synthetic speech augmentation. *arXiv preprint arXiv:1811.00707*, 2018.

Qi Li, Jinsong Zheng, Augustine Tsai, and Qiru Zhou. Robust endpoint detection and energy normalization for real-time speech and speaker recognition. *IEEE Transactions on Speech and Audio Processing*, 10(3): 146–157, 2002.

Fernando López and Jordi Luque. Iterative pseudo-forced alignment by acoustic ctc loss for self-supervised asr domain adaptation. *arXiv preprint arXiv:2210.15226*, 2022.

Zeping Min and Cheng Tai. Why pseudo label based algorithm is effective?–from the perspective of pseudo labeled data. *arXiv preprint arXiv:2211.10039*, 2022.

Zeping Min, Qian Ge, and Zhong Li. 10 hours data is all you need. *arXiv preprint arXiv:2210.13067*, 2022.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

Elena Rodríguez, Belén Ruíz, Ángel García-Crespo, and Fernando García. Speech/speaker recognition using a hmm/gmm hybrid model. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pp. 227–234. Springer, 1997.

Andrew Rosenberg, Yu Zhang, Bhuvana Ramabhadran, Ye Jia, Pedro Moreno, Yonghui Wu, and Zelin Wu. Speech recognition with augmented synthesized speech. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pp. 996–1002. IEEE, 2019.

Nick Rossenbach, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Generating synthetic audio data for attention-based speech recognition systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7069–7073. IEEE, 2020.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, Andrew Rosenberg, Bhuvana Ramabhadran, and Yonghui Wu. Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6699–6703. IEEE, 2020.

Youcef Tabet and Mohamed Boughazi. Speech synthesis techniques. a survey. In *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, pp. 67–70. IEEE, 2011.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 301–308. IEEE, 2017.

Sei Ueno, Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. Data augmentation for asr using tts via a discrete representation. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 68–75. IEEE, 2021.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11 (8):1240–1253, 2017.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015*, 2018.

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020.

Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Hannun, Gabriel Synnaeve, and Ronan Collobert. Iterative pseudo-labeling for speech recognition. *arXiv preprint arXiv:2005.09267*, 2020.

Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. *arXiv preprint arXiv:2102.01547*, 2021.

Cheng Yi, Jianzong Wang, Ning Cheng, Shiyu Zhou, and Bo Xu. Transfer ability of monolingual wav2vec2. 0 for low-resource speech recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2021.

Rui Zhao, Jian Xue, Jinyu Li, Wenning Wei, Lei He, and Yifan Gong. On addressing practical challenges for rnn-transducer. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 526–533. IEEE, 2021.