

# Multi-Level Modeling Units for End-to-End Mandarin Speech Recognition

Yuting Yang, Binbin Du, Yuke Li\*

NetEase Yidun AI Lab, Hangzhou, China

{yangyuting04, dubinbin, liyuke}@corp.netease.com

## Abstract

The choice of modeling units is crucial for automatic speech recognition (ASR) tasks. In mandarin scenarios, the Chinese characters represent meaning but are not directly related to the pronunciation. Thus only considering the writing of Chinese characters as modeling units is insufficient to capture speech features. In this paper, we present a novel method involves with multi-level modeling units, which integrates multi-level information for mandarin speech recognition. Specifically, the encoder block considers syllables as modeling units and the decoder block deals with character-level modeling units. To facilitate the incremental conversion from syllable features to character features, we design an auxiliary task that applies cross-entropy (CE) loss to intermediate decoder layers. During inference, the input feature sequences are converted into syllable sequences by the encoder block and then converted into Chinese characters by the decoder block. Experiments on the widely used AISHELL-1 [1] corpus demonstrate that our method achieves promising results with CER of 4.1%/4.6% and 4.6%/5.2%, using the Conformer and the Transformer backbones respectively.

**Index Terms:** multi-level modeling units, auxiliary task, mandarin speech recognition

## 1. Introduction

Speech recognition is a technology to translate human speech into textual representation. It is of importance for human-computer interaction, and is widely used in various scenarios. Conventional speech recognition systems consist of three separate components, including an acoustic model, a pronunciation model as well as a language model. The acoustic model calculates the probability distribution of each frame over the modeling units. The pronunciation model maps the modeling units, e.g., context-dependent states (CD-states) or context-dependent phonemes (CD-phonemes), into words. The language model scores different word combinations. These three components are trained separately. In the inference phase, the conventional ASR systems usually compose the phonetic model and the language model to build a decoding graph using a weighted finite state transducer (WFST) [2] and finally obtain the decoding results through the Viterbi algorithm [3]. In recent years, with the development of deep learning, end-to-end (E2E) ASR systems have become a hot spot both for industrial landing and academic research. End-to-end speech recognition systems merge the acoustic model, the pronunciation model, and the language model into a big neural network. Compared with the traditional speech recognition system, end-to-end ASR systems have the advantages of simpler architecture and better performance.

The choice of modeling units is related to the inherent characteristics of the language. English words usually consist of

several pronunciation units, while Chinese is a different scenario. Chinese characters are text symbols that are difficult to associate with pronunciation. Therefore, modeling with Chinese characters does not introduce phonological information into the network. On the other hand, Chinese has some unique characteristics, such as the pronunciation is toned and each Chinese character can be represented by a toned syllable (Pinyin with tone), which can be considered as pronunciation-related modeling units.

In this paper, we study how to organically combine both the syllable and the character modeling units in end-to-end ASR models. We hypothesize that both the phonological and the context-based information are important for system performance. To achieve this, we propose a novel method involves with multi-level modeling units, in which the encoder block considers syllables as modeling units and the decoder block deals with the character modeling units. The encoder block introduces the phonological information into the model by learning the alignment between the feature sequences and the syllable sequences, and the decoder block transduces syllable sequences into output characters. During inference, the input feature sequences are converted into syllable sequences by the encoder block and then converted into Chinese characters by the decoder block in a non-autoregressive way.

In this work, the inference is a two-pass process conducted by a unified end-to-end model. Specifically, the encoder module plays the role of converting feature sequences to syllable sequences. And the decoder plays the role of converting syllables to Chinese characters, which is a more challenging task compared with the character-to-character decoder module. For the above considerations, we introduce an auxiliary task termed *InterCE* to facilitate the conversion from syllable sequences to character sequences and improve the performance.

Our contributions are summarized as follows:

- We propose a novel end-to-end ASR model equipped with both the syllable-level and the character-level modeling units, which utilizes phonological as well as context-based information during the training process.
- We introduce an auxiliary task *InterCE* into the decoder block to further facilitate the conversion from syllable sequences to character sequences.
- We apply multi-level modeling units to Transformer [4] and Conformer [5] architectures respectively, and show the proposed method is effective for both architectures.

## 2. Related Work

### 2.1. E2E ASR

In recent years, end-to-end speech recognition (E2E ASR) systems have achieved outstanding improvements and become more and more popular. One research line is the connectionist temporal classification (CTC) [6] based methods. CTC-based

\* Corresponding author

ASR models calculate the probability distribution of each frame over the dictionary to obtain the alignment between the input and output sequences. In addition, recurrent neural network transducer (RNN-T) [7] is very popular in streaming on-device speech recognition since its low-latency and promising performance.

Another cluster is the attention-based models [4, 8, 9]. Transformer [4] is a dominant model among them, which has achieved state-of-the-art performance on neural machine translation (NMT). It adopts the encoder-decoder structure, both of which consist of several stacked sub-layers. Self-attention is the foundation module of Transformer, which is a mechanism linking all the position-pairs of a sequence to capture content-based global interactions. Transformer is also applicable to the task of speech recognition, in which the typical model is Speech-Transformer [10]. Speech-Transformer has been widely used in the field of ASR due to the nature of parallel training, the ability to capture long distance contextual information and its outstanding performance. Many efforts have been done on modifying the architecture of the Transformer to achieve better performance on speech recognition [5, 11]. For instance, Conformer [5] is a convolution-augmented Transformer, which combines Transformer with convolutional neural networks to extract both global and local dependencies of feature sequences.

## 2.2. E2E ASR with Auxiliary Tasks

There are lots of works that explored the improvements of auxiliary tasks for ASR based on the assumption that different layers learn representations at different levels [12–17]. Interlayer CTC [16] adopts the auxiliary CTC loss on the CTC-based model, it uses the intermediate representation of the encoder layer to calculate the auxiliary CTC loss. Auxiliary tasks are applied to the Transformer [15] and RNN-T networks [18] as well. In this work, We apply an auxiliary task to the decoder block to further enhance the ability of converting syllable representations to character representations. We study the improvements of different types of InterCE tasks under the multi-level modeling units framework in Section 4.

## 2.3. Modeling Units for Mandarin ASR

Many works have explored how to choose suitable modeling units for different end-to-end systems in mandarin speech recognition [19–22]. In [22], three modeling units achieve similar CER but syllable modeling units are slightly superior to character modeling units for the LSTM-CTC model, and the character-level modeling units are better than other options for attention-based models. In [20], five modeling units are compared based on Transformer structure, it draws conclusions that lexicon-free modeling units outperform lexicon-related modeling units and character is superior to other modeling units with Transformer models. All the above mentioned works show that character modeling units demonstrate good results for attention-based systems.

Many efforts have been made to exploit the pronunciation-related syllable modeling units for end-to-end systems, like the RNNT model [23] and the attention-based model [24]. They decouple the mandarin ASR task into two sub-tasks, including a recognition task converting the audio to syllables and a transcription task converting syllables to characters. In these cases, it is necessary to introduce an additional model to convert modeling units into Chinese characters, which makes the system more complicated and introduces cumulative errors into the system. In addition, syllables also served as modeling units

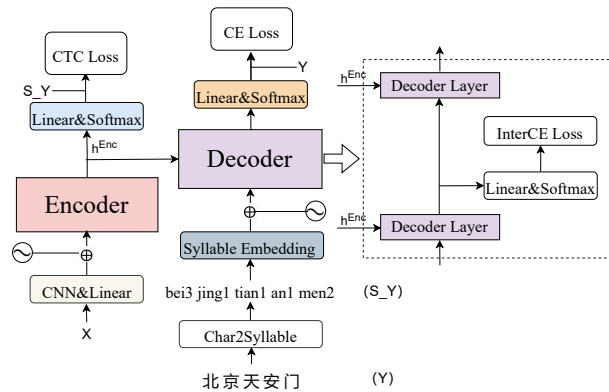


Figure 1: The framework with multi-level modeling units. ( $X$ ,  $S\_Y$ ,  $Y$ ) denote the input features, the syllable sequence and the character sequence respectively. We take InterCE loss for the output of some intermediate decoder layers. The InterCE auxiliary tasks only exist in some special decoder layers and are used only in the training process.

in the Mandarin-English code-switching ASR task [25].

In this paper, we propose a novel end-to-end mandarin speech recognition method equipped with multi-level modeling units, in which the pronunciation-related information is utilized by the model while learning the alignment between the feature sequences and the syllable sequences. Moreover, the inference process is conducted by a unified end-to-end model without introducing any additional models.

## 3. Proposed Method

We adopt two-level modeling units for the acoustic model, including character-level modeling units and syllable-level modeling units. We introduce the syllable-level modeling unit into the model to encode phonological information. At the same time, the character-level modeling units are retained to maintain the target character accuracy.

In the training process, we use the sequence of speech features  $X = \{x_1, x_2, x_3, \dots, x_T\}$  and the text sequence  $Y = \{y_1, y_2, y_3, \dots, y_n\}$  to train the network parameters. In this work, the syllable denotes Pinyin with tone. The syllable sequence  $S\_Y = \{sy_1, sy_2, sy_3, \dots, sy_n\}$  is obtained by an open source Chinese character to pinyin tool python-pinyin<sup>1</sup>. The given input  $X$  is a feature matrix with length  $T$  and dimension  $D$ . Each element  $y_i \in V_c$  is a token of the character sequence  $Y$  and each element  $sy_i \in V_s$  is a token of the syllable sequence  $S\_Y$ , where  $V_c$  is the character-level dictionary and  $V_s$  is the syllable-level dictionary. In the inference phase, we can obtain the text sequence  $Y$  given the speech feature sequence  $X$ .

### 3.1. Model Architecture

Figure 1 shows the framework with multi-level modeling units. We utilize a hybrid CTC/Attention method [26], in which the CTC classifier considers syllables as modeling units, and the decoder deals with character modeling units. We adopt the encoder block with Transformer as well as Conformer backbones. An encoder layer of Transformer is composed of two modules stacked together, i.e., a self-attention module and a feed-forward module. In contrast, an encoder layer of Con-

<sup>1</sup><https://github.com/mozillazg/python-pinyin>

Table 1: Character error rates with different  $\alpha$  and  $\beta$  values.

The value of $(\alpha, \beta)$	Dev set	Test set
$(\alpha=0.1, \beta=0.2)$	4.2	4.6
$(\alpha=0.1, \beta=0.4)$	4.1	4.6
$(\alpha=0.3, \beta=0.2)$	4.4	5.0
$(\alpha=0.3, \beta=0.4)$	4.4	4.9

former is composed of four modules, i.e., a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module. The decoder layer is composed of three modules stacked together, i.e., a masked self-attention module, a source-target multi-head attention module, and a feed-forward module. The encoder block consists of a stack of  $N$  identical layers and the decoder block consists of stacked  $L$  layers.

The main difference between the NMT Transformer [4] and the ASR Transformer [10] is the input of the encoder. The ASR Transformer utilizes several convolutional layers to produce the hidden representations, and a subsequent linear layer to map the dimension of flattened features to the attention dimension  $d_m$ . The outputs of the linear layer added with position encoding are fed to the encoder block. The encoder block extracts acoustic features represented  $h^{\text{Enc}}$  as

$$h^{\text{Conv}} = \text{Linear}(\text{Conv}(X)), \quad (1)$$

$$h^{\text{Enc}} = \text{Encoder}(h^{\text{Conv}}), \quad (2)$$

$$z^{\text{Enc}} = \text{Softmax}(\text{Linear}_{d_m \rightarrow |V_s|}(h^{\text{Enc}})), \quad (3)$$

$$\mathcal{L}_{\text{ctc}} = -\log P_{\text{ctc}}(S_Y | z^{\text{Enc}}), \quad (4)$$

where the outputs of the encoder block are applied to calculate syllable-level CTC loss. As shown in Eq.(3), the outputs of the encoder block are fed into a linear layer to map the dimension of  $h^{\text{Enc}}$  to syllable-level dictionary dimension  $|V_s|$ ,  $\text{Softmax}(\cdot)$  means a softmax function to get the posterior probability distribution over the syllable-level dictionary. As shown in Eq.(4), the outputs  $z^{\text{Enc}}$  together with the syllable labels  $S_Y$  are applied to get the CTC loss  $\mathcal{L}_{\text{ctc}}$ .

Given the acoustic feature representation  $h^{\text{Enc}}$ , the hidden states of hybrid modeling units can be formulated as

$$h_L^{\text{Dec}} = \text{Decoder}(h^{\text{Enc}}, h_0^{\text{Dec}}), \quad (5)$$

where  $h_0^{\text{Dec}}$  represents the *embedding* of the syllable sequence  $S_Y$  added with position encoding. The decoder uses the outputs of the encoder block and  $h_0^{\text{Dec}}$  to get the hidden representation  $h_L^{\text{Dec}}$ , where  $L$  is the number of decoder layers. According to this, the character-level CE loss is

$$c_L^{\text{Dec}} = \text{Softmax}(\text{Linear}_{d_m \rightarrow |V_c|}(h_L^{\text{Dec}})), \quad (6)$$

$$\mathcal{L}_{\text{att}} = -\log P_{\text{att}}(Y | c_L^{\text{Dec}}). \quad (7)$$

where the outputs of the decoder block are fed into a linear layer to map the dimension from  $d_m$  to  $|V_c|$  that is the size of the character-level dictionary. The final output  $c_L^{\text{Dec}}$  together with the character label  $Y$  are applied to get the CE loss  $\mathcal{L}_{\text{att}}$ .

### 3.1.1. InterCE Loss

We introduce interlayer auxiliary tasks to further facilitate the conversion from syllable sequences to character sequences. We call it *InterCE* loss here. The schematic diagram of the *InterCE*

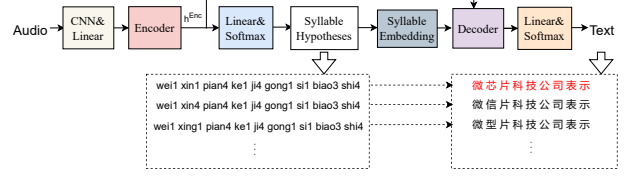


Figure 2: The pipeline of inference process.

is provided in Figure 1. We conduct the *InterCE* loss using the outputs of some special layers of the decoder block. Formally,

$$c_\epsilon^{\text{Dec}} = \text{Softmax}(\text{Linear}_{d_m \rightarrow |V_c|}(h_\epsilon^{\text{Dec}})), \quad (8)$$

$$\mathcal{L}_\epsilon = -\log P_{\text{att}}(Y' | c_\epsilon^{\text{Dec}}), \quad (9)$$

where  $\epsilon$  is a selected interlayer of decoder block.  $(|V_c|, Y') = (|V_c|, Y)$  when use the character-level auxiliary loss, and  $(|V_s|, Y') = (|V_s|, S_Y)$  with the syllable-level auxiliary loss.

$$\mathcal{L} = \alpha \mathcal{L}_{\text{ctc}} + \beta \frac{1}{k} \sum_{\epsilon \in \mathcal{R}} \mathcal{L}_\epsilon + (1 - \alpha - \beta) \mathcal{L}_{\text{att}} \quad (10)$$

When adding with *InterCE* loss in the model, the total loss function is a linear combination among the syllable-level CTC loss, the *InterCE* loss and the character-level CE loss. As shown in Eq.(10), where  $\mathcal{R}$  denotes the set of intermediate decoder layers, and  $k$  is the number of intermediate layers.

## 4. Experiments

We conduct experiments on AISHELL-1 [1] which is a benchmark dataset for mandarin ASR, containing 178 hours collected from 400 people. We apply speed perturbation [27] and SpecAugment [28] for training data. For all experiments, the speech features are 80-dimensional log-mel filterbank (Fbank) computed on 25ms window with 10ms shift. Table 2 describes the detailed composition of two-level modeling units in this study, the character dictionary consists of 4233 tokens and the syllable dictionary consists of 1370 tokens.

### 4.1. Experimental Setup

We follow the basic architectures of both Transformer [4] and Conformer [5]. The encoder is a 12-layer structure. The width of the attention layer is 256, the number of attention heads is 4, and the dimension in the feed-forward network sub-layer is 2048. The decoder is a 6-layer structure. Specifically, the kernel size is 15 for Conformer. The experiments are conducted using Wenet [29]. The Adam optimizer [31] with gradient clipping at 5.0 and warmup with 25000 steps are used during the training process. We employ dropout [32] to prevent over-fitting.

We conduct experiments to search for the best hyper-parameters of  $(\alpha, \beta)$ . The results are shown in Table 1. As for  $\alpha$ , we found that the performance is greatly improved with the decrease of  $\alpha$ . This might be because converting syllables to characters is a challenging task. As the  $\alpha$  decreases, the model pays more attention to enhancing conversion capabilities. As for  $\beta$ , the increase of  $\beta$  has little effect on performance improvement. We set  $\alpha = 0.1, \beta = 0.4$  in the following experiments.

We applied *InterCE* loss to different decoder layers. We use  $\mathcal{R} = \{3\}$  as *default* layer and  $\mathcal{R} = \{2, 4\}$  as *Multiple* layers. The different types of *InterCE* tasks are described as follows:

- **II**: Multiple layers with syllable-level *InterCE* loss.

Table 2: The descriptions of multi-level dictionaries.

Modeling units	Detailed composition	Examples
<i>Character-level</i>	4233 tokens including unknown token ⟨unk⟩ and start/end token ⟨sos/eos⟩	你好
<i>Syllable-level</i>	1370 tokens including ⟨blank⟩ token, ⟨unk⟩ token and ⟨sos/eos⟩ token	ni3 hao3

Table 3: Comparison of different models: *SUER* represents the syllable units error rate and *CER* represents the final character error rate. (I1,I2,I3,I4) denote different *InterCE* type.

Methods	Dev set		Test set	
	SUER	CER	SUER	CER
<b>Character-based</b>				
RNN-T [23]	-	-	-	6.1
Transformer [29]	-	4.8	-	5.3
<b>Syllable-based</b>				
Cascade-RNNT [23]	-	-	-	5.7
Decoupling+T [24]	-	-	4.3	6.1
Decoupling-T [24]	-	-	2.8	7.3
<b>Our work</b>				
Transformer + I1	2.7	4.7	3.3	5.3
Transformer + I2	2.9	4.8	3.4	5.3
<b>Transformer + I3</b>	2.7	<b>4.6</b>	3.2	<b>5.2</b>
Transformer + I4	2.9	4.8	3.4	5.2

Table 4: Performance with Conformer model. *SUER* represents the syllable units error rate and *CER* represents the final character error rate. (I1,I2,I3,I4) denote different *InterCE* type.

Methods	Dev set		Test set	
	SUER	CER	SUER	CER
<b>Character-based</b>				
Conformer + MMI [30]	-	4.5	-	4.9
Conformer	-	4.4	-	4.7
<b>Our work</b>				
Conformer + I1	2.2	4.1	2.6	4.6
Conformer + I2	2.2	4.1	2.6	4.6
Conformer + I3	2.2	4.1	2.6	4.7
<b>Conformer + I4</b>	2.1	<b>4.1</b>	2.5	<b>4.6</b>

- **I2**: Multiple layers with character-level *InterCE* loss.
- **I3**: Default layer with syllable-level *InterCE* losses.
- **I4**: Default layer with character-level *InterCE* loss.

#### 4.2. Inference

The pipeline of the inference process is shown in Figure 2. In the inference process, the encoder module extracts acoustic features using the input feature sequence  $X$ . The encoder module is followed by a linear projection layer and a softmax function to get the probability distribution over the syllable dictionary, which is applied to get syllable sequences using the CTC prefix beam search with beam size 10. Syllable sequences are converted into character sequences by the decoder block using greedy search. The final decoding results are sorted by the weighted sum of the CTC score and the attention score. This process is conducted in a non-autoregressive way.

#### 4.3. Main Results

The performance of the models was evaluated based on character error rate (CER) without using language models, and

Table 5: Ablation study of *InterCE* loss.

Model Architecture	Dev set	Test set
Character-based Conformer	4.4	4.7
+ character-level <i>InterCE</i> loss	4.3	4.7
Multi-level modeling units (I4)	4.1	4.6
- character-level <i>InterCE</i> loss	4.2	4.7

we show the syllable unit error rate (SUER) calculated by the syllable sequence for reference. we conducted experiments to compare end-to-end ASR models using different modeling units. The results of character-based Transformer and Conformer models are also obtained by the two-pass decoding method [29].

As shown in Table 3, we compare the multi-level modeling units method with character-only modeling units based Transformer [29] and RNN-T [23] networks. We can see that our proposed method achieves 4.6%/5.2% character error rate on the AISHELL-1 dev/test sets, which surpasses the character-based RNN-T [23] and Transformer [29] networks. Furthermore, we compared our method with the models using syllable modeling units, and the proposed method outperforms the cascade RNN-T [23] and decoupling models [24] as well.

The proposed method also achieves good results with the Conformer network. As shown in Table 4, when applied to the Conformer architecture, our method achieves CER of 4.1%/4.6% on the dev/test sets. It outperforms the character-based Conformer model as well as recently emerged methods using the Conformer backbone [30, 33].

#### 4.4. Ablation Study

We study the effects of *InterCE* loss for the final performance using the Conformer network. First, we study adding character-level *InterCE* loss to character-only based conformer network, and see a slight improvement in the dev dataset. Second, we try removing *InterCE* loss from our multi-level modeling units framework and find that it degrades the results by 0.1 on dev/test sets. These results in Table 5 suggest the advantage of unifying syllable-level modeling units and character-level modeling units in an end-to-end model. And the *InterCE* loss makes additional improvements under this multi-level modeling units framework.

## 5. Conclusions

In this paper, we propose a novel end-to-end mandarin speech recognition method equipped with multi-level modeling units, which allows the model to integrate multi-level information. During the inference process, the input feature sequences are converted into syllable sequences and subsequently converted into Chinese characters within a unified end-to-end model. We further utilize the *InterCE* loss to improve the accuracy of converting syllables to Chinese characters. The proposed model achieves competitive results and outperforms recently published literature on the AISHELL-1 benchmark.

## 6. References

- [1] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.
- [2] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.
- [3] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [5] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.
- [6] A. Graves, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *Proc. Int. Conf. on Machine Learning, 2006*, 2006.
- [7] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
- [8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.
- [9] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Computer ence*, vol. 10, no. 4, pp. 429–439, 2015.
- [10] L. Dong, S. Xu, and B. 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)*. IEEE, 2018, pp. 5884–5888.
- [11] A. Mohamed, D. Okhonko, and L. Zettlemoyer, "Transformers with convolutional context for asr," *arXiv preprint arXiv:1904.11660*, 2019.
- [12] K. Krishna, S. Toshiwal, and K. Livescu, "Hierarchical multitask learning for ctc-based speech recognition," *arXiv preprint arXiv:1807.06234*, 2018.
- [13] R. Sanabria and F. Metzke, "Hierarchical multitask learning with ctc," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 485–490.
- [14] S. Toshiwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," *arXiv preprint arXiv:1704.01631*, 2017.
- [15] A. Tjandra, C. Liu, F. Zhang, X. Zhang, Y. Wang, G. Synnaeve, S. Nakamura, and G. Zweig, "Deja-vu: Double feature presentation and iterated loss in deep transformer networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6899–6903.
- [16] J. Lee and S. Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6224–6228.
- [17] J. Nozaki and T. Komatsu, "Relaxing the conditional independence assumption of ctc-based asr by conditioning on intermediate predictions," *arXiv preprint arXiv:2104.02724*, 2021.
- [18] C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig, "Improving rnn transducer based asr with auxiliary tasks," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 172–179.
- [19] S. Zhang, M. Lei, Y. Liu, and W. Li, "Investigation of modeling units for mandarin speech recognition using dfsmn-ctc-smbt," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7085–7089.
- [20] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 210–220.
- [21] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," *arXiv preprint arXiv:1804.10752*, 2018.
- [22] W. Zou, D. Jiang, S. Zhao, G. Yang, and X. Li, "Comparable study of modeling units for end-to-end mandarin speech recognition," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 369–373.
- [23] X. Wang, Z. Yao, X. Shi, and L. Xie, "Cascade rnn-transducer: Syllable based streaming on-device mandarin speech recognition with a syllable-to-character converter," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 15–21.
- [24] J. Yuan, X. Cai, D. Gao, R. Zheng, L. Huang, and K. Church, "Decoupling recognition and transcription in mandarin asr," *arXiv preprint arXiv:2108.01129*, 2021.
- [25] S.-P. Chuang, H.-J. Chang, S.-F. Huang, and H.-y. Lee, "Non-autoregressive mandarin-english code-switching speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 465–472.
- [26] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [27] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.
- [28] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," 2019.
- [29] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, "Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," in *Proc. Interspeech*. Brno, Czech Republic: IEEE, 2021.
- [30] J. Tian, J. Yu, C. Weng, S.-X. Zhang, D. Su, D. Yu, and Y. Zou, "Consistent training and decoding for end-to-end speech recognition using lattice-free mmi," *arXiv preprint arXiv:2112.02498*, 2021.
- [31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [33] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.