Hybrid CTC Language Identification Structure for Mandarin-English Code-Switching ASR

Hengxin Yin[1], *Guangyu Hu*[1], *Fei Wang*[1], *Pengfei Ren*[2]

[1]Didi Chuxing, [2]Xi'an Jiaotong University

[1]{hengxinyin, huguangyu, warrenwangfei}@didiglobal.com,
[2]pengfei.ren@stu.xjtu.edu.cn

Abstract

With the advent of globalization, there is a growing demand for code-switching automatic speech recognition (ASR), which can accurately discriminate a speaker who alternates words of two or more languages within a single sentence or across sentences. The common phenomenon of Mandarin-English code-switching ASR is quite common around the world, like China and Singapore.

In this paper, we propose a hybrid CTC language identify-cation architecture and a complete model training method for Mandarin-English code-switching ASR. We propose an ASR architecture that adds a LID layer after CTC decoder which based on CTC-AED architecture to calculate the CTC LID loss at the frame level. Furthermore, a fusion loss based on CTC loss, Attention loss and CTC LID loss is used to train Mandarin-English code-switching ASR model. In order to enable the ASR model to quickly converge and strengthen the language identification ability, a dynamic CTC LID loss weight is introduced.

Results show that the proposed model architecture and model training method can effectively improve performance of the Mandarin-English code-switching ASR model. The final system achieves a Mixed Error Rate (MER) of 20.9% in the ISCSLP2022 Magichub Code-Switching ASR Challenge.

Index Terms: speech recognition, code-switching, language identification loss

1. Introduction

Code-switching (CS) is defined as the language alternation in an utterance. Recognizing CS speech is essential because it is frequently used in everyday conversations in multilingual regions like Singapore, Malaysia and Hong Kong [1]. However, the majority of commercial ASR systems are only intended to recognize one language, which restricts the applications. Building a code-switching speech recognition system is much more difficult than building a monolingual speech recognition system.

Recently, automatic speech recognition (ASR) has a significant trend from hybrid modeling [2] based on deep neural networks to end-to-end (E2E) [3-6]. Because E2E models achieve the state-of-the-art results in most benchmarks in terms of ASR and it is easier to directly translates an input speech sequence into an output token sequence using a single network. The most popular E2E techniques for ASR are: 1) Connectionist Temporal Classification (CTC) [7], CTC aims to map speech input sequences to output label sequences. Since the length of the output label is smaller than the length of the input speech frame, blank labels are inserted between the output labels that allow duplication to build a CTC path with the same length as the input speech frame. 2) Recurrent neural network Transducer (RNN-T) [8-10]. RNN-T has an encoder network, a prediction network, and a joint network. RNN-T removes the conditional independence assumption of CTC, provides a natural way for streaming ASR. 3) Attention-based Encoder-Decode (AED) [11-14]. AED contains an encoder network, an attention module, and a decoder network. Additionally, joint CTC-attention models [15] take advantage of both the CTC and sequence-to-sequence models' advantages within the context of multi-task learning, which improves performance and robustness.

Data shortage is one of the major issues for CS ASR task. E2E ASR needs a lot of data to train models, and data collecting is time and money consuming. Several attempts have been made to alleviate the CS data scarcity problem. [16-17] used the Text-To-Speech (TTS) technology to produce new speech that corresponded to CS texts. [18-19] used un-transcribed CS voice data through the application of semi-supervised methods. [20-22] employed transfer learning techniques train the CS models. [16] separated the language-dependent segments into individual pieces, then randomly spliced to create new CS utterances. These techniques produced notable improvements due to the increased training data.

In this paper, the ISCSLP2022 Magichub Code-Switching ASR Challenge [35] provides approximately 555.9 hours of Mandarin-English CS data and 150 hours of Mandarin data. Using these data, we propose a hybrid CTC language identification structure and a complete model training method for Mandarin-English code-switching ASR.

2. Methods

2.1. Model architecture

To train the Mandarin-English CS models, we use an end-to-end model structure, as shown in Figure 1. The model structure consists of four parts: Shared Encoder, Attention Decoder, CTC Decoder and Language Identification (LID) Layer.

The Shared Encoder mainly encodes input audio acoustic features X to generate latent representation h, which contains many potential information of speech.

$$h = Shared\ Encoder(X) \tag{1}$$

The Attention Decoder and the CTC Decoder are used to decode the latent representation obtained from the output of the Shared Encoder in their respective ways to get the final recognition result.

$$P_{CTC}(\mathrm{Y}\mid \mathrm{X}) = CTC\ decoder(h) \tag{2}$$

$$P_{att}(Y_t \mid X) = Attention\ decoder(h, Y_{:t-1}) \tag{3}$$

The LID Layer is attached to the CTC Decoder and is used to map the output tokens of the CTC Decoder to the corresponding language tokens.

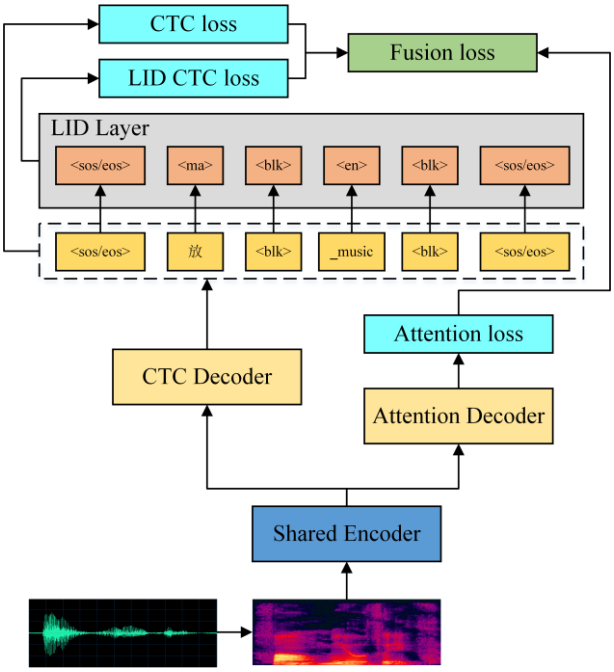

Figure 1: *The end-to-end model structure.*

2.2. Acoustic modeling unit

The choice of the model unit is a crucial component of E2E modeling. For Mandarin ASR, the acoustic modeling units usually use phone [23], cd-phone [24], syllable or characters [25-26]. Meanwhile, words [27-28], subwords, letters are for English ASR. Combining Chinese characters with English letters to create new modeling units in Mandarin-English code-switching task [29].

We use BPE subwords instead of letters as English modeling units. Three reasons are as follows: 1) the numbers of Chinese characters are more than English letters, using subwords will reduce the modeling units gap between Mandarin and English. 2) the acoustic counterpart of English letter is shorter than Chinese character. 3) compared to word units, the BPE subwords units is demonstrated to both improve performance and address the OOV issue for E2E English ASR [30].

As a result, in this work, we use BPE subwords as the acoustic modeling unit for English and Chinese characters for Mandarin. Besides, we still use three additional units, namely <unk>, <blk> and <sos/eos>. The <unk> represents an unknown character or subword, the <blk> represents the extra blank label of CTC, and the <sos/eos> represents the beginning and end of an audio.

2.3. Training loss

The pronunciation of Mandarin is different from that of English, and the duration of pronunciation of a Mandarin character and a word is also different. For example, the pronunciation of a Mandarin character is usually one syllable, while the pronunciation of an English word may have multiple syllables. This results in low recognition accuracy of CTC Decoder for Mandarin-English code-switching ASR.

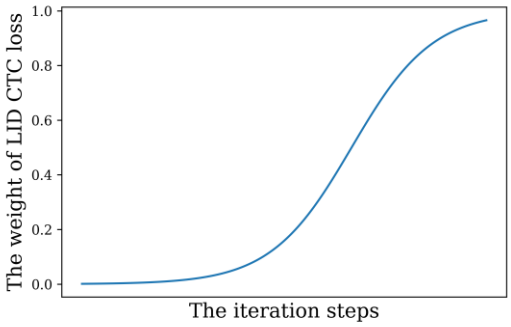

Figure 2: *The curve of the LID CTC loss weight increasing with the number of iteration steps.*

In order to improve the accuracy of language identification, without increasing the number of model parameters, we add the LID Layer after the CTC Decoder module and calculate the Language CTC loss (LID CTC loss). First, we map the *softmax* value $P \in \mathbb{R}^{n\times m}$ (n: the length of acoustic modeling unit. m: the audio frames) output from the CTC Decoder module to $P' \in \mathbb{R}^{5\times m}$ containing only five types of units: <blank>, <unk>, <ma>, <en>, <sos/eos>. The mapping function as follows:

$$p_i' = \begin{cases} p_i,\ i \in [\text{<blank>, <unk>,<sos/eos>}] \\ \max(p_{mandarin}) \quad ,\ i == < ma > \\ \max(p_{english}) \quad ,\ i == < en > \end{cases} \tag{4}$$

where the <ma> indicates that the frame is identified as a Mandarin unit, and the <en> indicates that the frame is identified as an English unit.

The values of <blank>, <unk>, <sos/eos> after mapping are the same as those before mapping. The values of <ma> and <en> are respectively the Mandarin and English unit with the largest *softmax* value in the corresponding frame.

The LID CTC loss is calculated by $P' \in \mathbb{R}^{5\times m}$ obtained from the mapping relationship. And use fusion loss to train the model. The fusion loss can be calculated by the following formula:

$$loss = \lambda loss_{CTC} + (1-\lambda) loss_{att} + \alpha loss_{LID-\mathrm{CTC}} \tag{5}$$

where $loss_{CTC}$ is the CTC loss, $loss_{Att}$ is the Attention loss, $loss_{LID-CTC}$ is the LID CTC loss, λ is the CTC loss weight, which is generally set to 0.5, α is the LID CTC loss weight.

In order to make the training converge quickly and learn the language information more accurately after convergence, it is inappropriate to take a constant as the LID CTC loss weight. It is more reasonable to set the value of α to increase as the number of iteration steps increases. We draw lessons from the *sigmoid* function and build $\alpha = f(step)$ function as follows:

$$\alpha = f(steps) = \frac{1}{1+\exp(-\frac{steps - S}{1.5(S\times 10)})} \tag{6}$$

where *steps* is the number of real-time iteration steps in the model training process, and *S* is the final iteration steps in the whole training process. The calculation method is as follows:

$$S = \text{total epoch} \times \frac{\text{sample size}}{\text{batch size}} \tag{7}$$

The curve of the LID CTC loss weight α increasing with the number of iteration steps is shown in Figure 2. The value of α is always between 0 and 1.

3. Experiments

The audio data are sampled at 16 kHz. Use 80 dimensions FBank as data feature. The Mandarin-English code-switching ASR models are trained based on WeNet Toolkit [31].

3.1. Data

This paper only uses the data within the requirements of the ISCSLP2022 Magichub Code-Switching ASR Challenge [32] to train the acoustic model. The details of the train set, development set (dev set) and test set are shown in Table 1.

Two open-source datasets are used to train the model, namely TALCS and MagicData-RAMC. The TALCS corpus [33] contains 587 hours of Mandarin-English code-switching speech data recorded from real online one-to-one English teaching scenes with a sampling rate of 16 kHz. The MagicData-RAMC [34] is a corpus of conversational speech data recorded from native speakers of Mandarin Chinese over mobile phones，which contains roughly 180 hours of speech sampled at 16 kHz. A total of 705.55 hours of train sets from the TALCS and the MagicData-RAMC are used to train the speech recognition models.

The development set and test set provided by the ISCSLP2022 Magichub Code-Switching ASR Challenge are used to evaluate the performance of the Mandarin-English code-switching ASR models.

Before model training, the data of train set and dev set need to be cleaned. Transcriptions with some special symbols and invalid audio are removed. At the same time, the audio with [+] and [*] symbols in the transcriptions of the train set and dev set is deleted. The transcriptions with [LAUGH-TER], [SONANT], [MUSIC] symbols of audio are screened as noise for data augmentation.

Table 1: *Data description for train, dev and test set.*

Datasets	Source (language)	Duration (hours)	Number of sentences
Train set	TALCS (ma-en)	555.9	350000
	MagicData-RAMC (ma)	149.65	219325
Dev set	Dev (ma-en)	3.5	6330
Test set	Test (ma-en)	6.8	11243

We generate 5000 subwords units from Librispeech using BPE, and 4324 Chinese characters are extracted from train set. There are totally 9327 tokens for modeling (with three extra tokens for <unk>, <blank> and <sos/eos>).

3.2. Data augmentation

We perform a series of data augmentation work on the train set, including noise insertion, speed perturbation and SpecAugment [35].

Noise insertion: the text filtered from the train set containing special symbols such as [LAUGHTER], [SONANT] and [MUSIC] is regarded as noise data. We also use open-source noise data MUSAN (OpenSLR17). For each audio in the train set, one noise in the noise set is randomly selected, and the Signal-to-Noise Ratio (SNR) of each audio is randomly set in the range of 10 to 30.

Speed perturbation: for each audio in the train set, 0.9, 1.0, 1.1 times of speech speed perturbation is applied to enhance the generalization ability of the model for audio with different speech speeds.

SpecAugment: the SpecAugment method proposed in [35] is used to mask the time domain and frequency domain part in the training process, and the SpecAugment parameters *num_t_mask* is set as 2, *num_f_mask* as 2, *max_t* as 50, *max_f* as 10, *max_w* as 80.

3.3. Model training

In this paper, the Shared Encoder use multiple Conformer layers. As for the decoders, the multiple BiTransformer layers in WeNet are used for the Attention Decoder and a linear layer for CTC Decoder. The model parameters are shown in Table 2.

Table 2: *Parameter description of the code-switching ASR model.*

Parameters	Shared Encoder (Conformer)	Attention Decoder (BiTransformer)	CTC Decoder
output_size	512	9327	9327
attention_heads	8	8	-
linear_units	2048	2048	-
num_blocks	12	3	-
dropout_rate	0.1	0.1	-
positional_dropout_rate	0.1	0.1	-
attention_dropout_rate	0.1	-	-
cnn_module_kernel	15	-	-
r_num_block	-	3	-

We set the following models for comparison to highlight the advantages of our models.

Base: use the basic E2E model. The Shared Encoder and the decoders use multiple Conformer layers, multiple BiTransformer layers and CTC linear layer mentioned above in this paper. The original train set is used to train the model. The loss calculation method does not involve the LID CTC loss. The calculation formula of loss as follows:

$$loss = \lambda loss_{CTC} + (1-\lambda) loss_{att} \tag{8}$$

where the weight of CTC loss λ is set as 0.5.

Base+DA: based on the Base model, the train set is augmented with three methods: noise insertion, speed perturbation and SpecAugment. The details are covered in Section 3.2.

Base+DA+LID: on the basis of Base+DA model, the LID CTC loss proposed in this paper has been used. In the process of model training, CTC loss, attention loss and LID CTC loss are integrated. The details of fusion loss are elaborated in Section 2.3.

Table 3: *The MERs of dev set and test set (%).*

Datasets	Models	Decoding models			
		ctc_greedy_search	ctc_prefix_beam_search	attention	attention_rescoring
dev	Base	25.8	25.8	24.9	24.3
	Base+DA	25.2	25.3	24.4	23.7
	Base+DA+LID	23.6	23.6	22.9	**22.3**
test	Base	24.2	24.5	24.5	23.2
	Base+DA	24.0	24.1	24.0	22.6
	Base+DA+LID	22.7	22.7	22.7	**21.5**

All models are train 70 epochs. Additionally, we create our final model by averaging the top-10 models that perform the best and have the lowest loss on the dev set during training. We use four decoding models (ctc_greedy_search, ctc_prefix_ beam_search, attention, attention_rescoring) mentioned in [15] to obtain model recognition results respectively.

3.4. Language model

In order to obtain more accurate Mandarin-English code-switching recognition results, we train the N-gram language model for shallow fusion. We use the train set text, Chinese and open-source English corpus to train the 4-gram model.

3.5. Evaluation measures

For different modeling units and language scenarios, the common evaluation indicators include the Word Error Rate (WER) and the Character Error Rate (CER). For the Mandarin-English code-switching ASR model, we need to calculate both the CER for Mandarin and the WER for English. Therefore, this paper uses the Mixed Error Rate (MER) as the evaluation index. *Sclite*, an evaluation open-source tool, is used for scoring.

4. Results and Discussion

Table 3 shows the MERs of different models for the identify-cation results of dev set and test set. We use ctc_greedy_ search, ctc_prefix_beam_search, attention and attention_re-scoring decoding models respectively to obtain the results. The analysis results show that:

1. With data augmentation for the train set, the MERs for both dev set and test set is reduced by 0.6% on attention_rescoring. The MERs decrease in the range of 0.2% to 0.6% for the different decoding models.

2. The model trained with fusion loss proposed in this paper has a greater reduction in MER. The MERs for dev set and test set decreased by 1.4% and 1.1%, respectively, on attention_ rescoring.

3. Since the proposed LID CTC loss is implemented through the CTC Decoder, the MERs reduction is more obvious on the two decoding models ctc_greedy_search and ctc_prefix_ beam_search.

In order to further analyze the performance improvement details of the Base+DA+LID model. We analyze the MERs of substitution, deletion and insertion of the results, as shown in Table 4. It can be analyzed that the Base+DA+LID model can mainly reduce the MERs of substitution, deletion, especially the MER of substitution. The MERs of substitution is reduced by 1.0% and 0.8% on the dev set and test set,

Table 4: *The MERs of substitution, deletion and insertion on attention_rescoring.*

Datasets	Models	SUB	DEL	INS	Total MER
dev	Base+DA	17.4	2.9	3.4	23.7
	Base+DA +LID	16.4	2.4	3.5	22.3
	difference	-1.0	-0.5	+0.1	-1.4
test	Base+DA	16.7	2.6	3.4	22.6
	Base+DA +LID	15.9	2.2	3.4	21.5
	difference	-0.8	-0.4	0	-1.1

respectively. Thus, adding LID layer can effectively improve the performance for Mandarin-English code-switching ASR.

In order to further reduce the substitution errors, we use the 4-gram language model to rescore the recognition results of the Base+DA+LID model. Due to the small proportion of Mandarin-English code-switching corpus expected in the language model training, we need to reduce the rescore weight of the language model, and the final acoustic scale is set to 3.5

Table 5 shows MERs after shallow fusion using the language model. The result shows that adding the language model significantly reduces the MER of substitution, with 2.2% reduction in the dev set and 1.1% reduction in the test set. The MER of test set was reduced to 20.9%.

Table 5: *The MER comparison after shallow fusion using the language model*.

Datasets	Models	SUB	DEL	INS	Total MER
dev	Base+DA+ LID	16.4	2.4	3.5	22.3
	Base+DA+ LID+LM	14.2	3.4	2.7	20.3
	difference	-2.2	+1.0	-1.2	-2.0
test	Base+DA+ LID	15.9	2.2	3.4	21.5
	Base+DA+ LID+LM	14.8	3.7	2.5	20.9
	difference	-1.1	+0.5	-0.9	-0.6

5. Conclusions

In this work, we proposed a hybrid CTC language identify-cation structure for Mandarin-English code-switching ASR. The model structure is improved based on CTC-AED structure, and LID layer is added after CTC Decoder to judge the language at frame level. It can enhance the language identification ability of the model without increasing the

number of model parameters. At the same time, the dynamic CTC LID loss weight is introduced, so that the model can gradually strengthen the discrimination ability of languages while rapidly converging. This paper also describes the details of the whole construction model. Experimental results show that our proposed structure and model training method is effective for Mandarin-English Code-Switching ASR.

6. References

[1] D. C. Li, "Cantonese-English code-switching research in Hong Kong: a Y2K review," *World Englishes*, vol. 19, no. 3, pp. 305–322, 2000.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012

[3] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-End continuous speech recognition using attention-based recurrent NN: First results," *arXiv preprint arXiv:1412.1602*, 2014.

[4] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding," in *Proc. ASRU. IEEE*, 2015, pp. 167–174.

[5] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP*, 2016, pp. 4945–4949.

[6] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Interspeech*, 2017, pp. 939–943.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[8] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[9] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP*, 2020, pp. 6059–6063

[10] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao et al., "Developing RNN-T models surpassing high-performance hybrid models with customization capability," in *Interspeech*, 2020, pp. 3590–3594.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[13] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577– 585.

[14] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016, pp. 4960–4964.

[15] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, X. Lei, "WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," in *Interspeech*, 2021, pp. 4054–4058.

[16] C. Du, H. Li, Y. Lu, L. Wang, and Y. Qian, "Data augmentation for end-to-end code-switching speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT). IEEE*, 2021, pp. 194–200.

[17] S. Nakayama, A. Tjandra, S. Sakti, and S. Nakamura, "Speech chain for semi-supervised learning of Japanese-English code-switching ASR and TTS," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 182–189.

[18] E. Yimaz, M. McLaren, H. van den Heuvel, and D. A. van Leeuwen, "Semi-supervised acoustic model training for speech with code-switching," *Speech Communication*, vol. 105, pp. 12–22, 2018.

[19] P. Guo, H. Xu, L. Xie, and E. S. Chng, "Study of semi-supervised approaches to improving English-Mandarin code-switching speech recognition," in *Interspeech*, 2018, pp. 1928–1932.

[20] N. Luo, D. Jiang, S. Zhao, C. Gong, W. Zou, and X. Li, "Towards end-to-end code-switching speech recognition," *arXiv preprint arXiv:1810.13091*, 2018.

[21] K. Li, J. Li, G. Ye, R. Zhao, and Y. Gong, "Towards code-switching ASR for end-to-end CTC models," in *ICASSP*, 2019, pp. 6076–6080

[22] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, "Investigating end-to-end speech recognition for Mandarin-English code-switching," *in ICASSP*, 2019, pp. 6056–6060.

[23] T. N. Sainath, R. Prabhavalkar, S. Kumar, S. Lee, A. Kannan, D. Rybach, V. Schogol, P. Nguyen, B. Li, Y. Wu et al., "No need for a lexicon? evaluating the value of the pronunciation lexica in end-to-end models," *in ICASSP*, 2018, pp. 5859–5863.

[24] H. Sak, A. Senior, K. Rao, et al. "Fast and accurate recurrent neural network acoustic models for speech recognition," *arXiv preprint arXiv:1507.06947*, 2015.

[25] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," *in International conference on machine learning*, 2014, pp. 1764–1772.

[26] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[27] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[28] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for English conversational speech recognition," *in ICASSP*, 2018, pp. 4759–4763.

[29] H. Seki, S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "An end-to-end language-tracking speech recognizer for mixed-language speech," *in ICASSP, 2018*.

[30] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina et al., "State-of-the-art speech recognition with sequence-to-sequence models," *in ICASSP, 2018*.

[31] B. Zhang, D. Wu, Z. Peng, et al. "WeNet 2.0: More Productive End-to-End Speech Recognition Toolkit," *arXiv preprint arXiv: 2203.15455*, 2022.

[32] "ISCSLP2022 Magichub Code-Switching ASR Challenge," https://magichub.com/competition/code-switching-asr-challenge/?nocache, 09 2022, (Accessed 07/10/2022 23:00).

[33] C. Li, S. Deng, Y. Wang, G. Wang, Y. Gong, C. Chen, J. Bai, "TALCS: An open-source Mandarin-English code-switching corpus and a speech recognition baseline," in *Interspeech*, 2022, pp. 1741–1745

[34] Z. Yang, Y. Chen, L. Luo, R. Yang, L. Ye, G. Cheng, J. Xu, Y. Jin, Q. Zhang, P. Zhang, et al., "Open Source MagicData-RAMC: A Rich Annotated Mandarin Conversational (RAMC) Speech Dataset," *arXiv preprint arXiv:2203.16844*, 2022.

[35] 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," *arXiv preprint arXiv:1904.08779*, 2019.