INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

ECTC-DOCD: An End-to-end Structure with CTC Encoder and OCD Decoder for Speech Recognition

Cheng Yi[1,2], *Feng Wang*[1], *Bo Xu*[1]

[1]Institute of Automation, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
{yicheng2016,feng.wang,xubo}@ia.ac.cn

Abstract

Real-time streaming speech recognition is required by most applications for a nice interactive experience. To naturally support online recognition, a common strategy used in recently proposed end-to-end models is to introduce a blank label to the label set and instead output alignments. However, generating the alignment means decoding much longer than the length of the linguistic sequence. Besides, there exist several blank labels between two output units in the alignment, which hinders models from learning the adjacent dependency of units in the target sequence. In this work, we propose an innovative encoder-decoder structure, called ECTC-DOCD, for online speech recognition which directly predicts the linguistic sequence without blank labels. Apart from the encoder and decoder structures, ECTC-DOCD contains an additional shrinking layer to drop the redundant acoustic information. This layer serves as a bridge connecting acoustic representation and linguistic modelling parts. Through experiments, we confirm that ECTC-DOCD can obtain better performance than a strong CTC model in online ASR tasks. We also show that ECTC-DOCD can achieve promising results on both Mandarin and English ASR datasets with first and second pass decoding.

Index Terms: end-to-end, streaming ASR, encoder-decoder, OCD, CTC

1. Introduction

The speech recognition task involves transforming a long acoustic feature sequence to a short label sequence. Emerging end-to-end models have better performance than traditional models in ASR task over many datasets [1, 2, 3, 4]. Besides the competing performance, a much simpler process and negligibly required expert knowledge make the end-to-end modelling approach attractive.

There are two mainstream branches to handle this transformation applied in end-to-end models: 1) referring to the whole acoustic information by attention mechanism [5] during decoding, including Listen, Attend and Spell (LAS) [6] and Joint CTC-Attention [1]; 2) adding an additional blank label to the output label set, degenerating ASR task to sequence labeling task. Typical models contain Connectionist Temporal Classification (CTC) [7, 2], Recurrent Neural Aligner (RNA) [8] and Recurrent Neural Network Transducer (RNN-T) [9]. The former strategy requires the entire input sequence accessible in advance at each decoding step, thus cannot be directly used for real-time streaming speech recognition. The latter way promotes the model to generate the alignment of the input sequence and subsequently map the alignment into the linguistic sequence. It is naturally suitable for the model to work in a streaming way, and we focus on this approach.

However, generating an alignment requires the model to decode as long as the acoustic feature sequence, which is much longer than the target sequence in most cases. Since the blank and repeated labels in the alignment will be removed, it is a waste of time decoding on these steps. Besides the redundant computation, these labels also cause trouble for the model to learn the dependency between the adjacent units in the target sequence, which is the key to model the language [10].

In this work, we apply an encoder-decoder structure [11] to process frames coming streamingly and generate the linguistic sequence directly. The encoder is supervised to generate a hidden representation sequence for the input feature with CTC loss. Then this representation sequence is shrunk to feed the decoder by merging the repeated frames and removing the ones corresponding to blank labels (called *blank frames*). The decoder outputs the final posterior probability distributions which have the same length as the shrunk hidden representation. Cross-Entropy (CE) [12] loss is applied to the distributions according to the target sequence. To solve the length mismatch between distributions and target sequence when computing the CE loss, Optimal Completion Distillation (OCD) [13] is used to select the target label(s) at each step of decoding. Due to the supervisions used in encoder and decoder, we call the structure ECTC-DOCD.

The key contributions of our model include:

- We fuse the pioneering shrinking layer in structure to skip the blank frames in the encoded representation, bridging the length gap of acoustic and linguistic parts. This layer can relief the redundant computation on blank and repeated frames, thus the model can directly output linguistic sequences without blank labels.
- Two different tasks in ECTC-DOCD make it partly explainable: The CTC loss function forces the model to learn which frames belong to the blank and repeated labels; The CE loss augmented by the OCD pushes the model to generate optimal labels by additionally referring to the partly decoded labels.
- We propose a matched learning paradigm for the two-level supervision task in ECTC-DOCD, where the model's focus of learning gradually switches from one to the other.

2. Related Works

Starting from the pioneer CTC work [14], the extension of the output set with a blank label transforms the task of generating a linguistic sentence to generating an alignment, making

This work is supported by the National Key Research and Development Program of China under No.2016YFB1001404.

http://dx.doi.org/10.21437/Interspeech.2019-1212

it an end-to-end way to realise the streaming speech recognition. Unfortunately, traditional CTC models make a conditional independence assumption for label predictions [8, 9], thus unable to learn an internal language model (LM) for the output sequences.

To leverage additional language information for CTC, RNN-T [9] introduces a transcription network trained with text data and marginalizes all the legal alignments within an output lattice. Different from RNN-T, ECTC-DOCD utilizes the decoder under the encoder-decoder structure, similar to the mainstream seq2seq models. Moreover, the decoder in ECTC-DOCD can learn the label dependency during training from scratch.

From the viewpoint of model structure, our model is much like RNA, which can also process stream-coming frames. They have the same working flow except for the shrinking layer between the encoder and decoder. As a result, RNA outputs alignment and is trained with a CTC-like loss. Without the interruption of blank labels, ECTC-DOCD has better language modelling ability than RNA theoretically.

The shrinking idea is introduced in [15] for the first time as we know. That work skips the search of blank-dominated steps during CTC decoding, losing no accuracy but obtaining 2-3 times speedup. Our shrink layer is equivalent to omitting the blank frames, condensing the decoding steps.

A previous work [16] also suggests using different level supervisions for end-to-end models, which belongs to multitask learning [17, 18]. In multitask learning, however, there is only one primary task. The rests are auxiliary, and they are not necessary for the primary one. In contrast, two tasks are cascaded in ECTC-DOCD, breaking down the whole ASR task into acoustic representation and linguistic generation.

3. Model Description

ECTC-DOCD is a neural network within the encoder-decoder framework that models the mapping between input and output sequences in an end-to-end way. Given the acoustic feature sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$ with length T and the alphabet of labels $\mathscr{C}$, ECTC-DOCD outputs the predicted label sequence $\mathbf{z} = (z_1, z_2, ..., z_U)$ with length U, where $z_i \in \mathscr{C}$.

3.1. Encoder with CTC loss

As demonstrated in Figure 1, the encoder outputs the hidden representation sequence, which is then fed into a projection and softmax layer to generate a distribution sequence over $\mathscr{C}'$. Here $\mathscr{C}'$ is the label set extended with a blank label ϵ, i. e. $\mathscr{C}' = \mathscr{C} \cup \epsilon$, and CTC loss is applied to the distribution sequence. The label sequence output by the encoder can be viewed as the alignment. The computations in the encoder part are formally described as:

$$\mathbf{h} \triangleq \text{encoder}(\mathbf{x}) \tag{1}$$

$$p_{\text{en}} = \text{softmax}(\mathbf{W} * \mathbf{h}) \tag{2}$$

$$\mathbf{h}' = \text{shrink}(\mathbf{h}, \arg\max(p_{\text{en}})) \tag{3}$$

The shrinking function merges the repeated frames and filters out the blank frames. The detailed operations are shown in Figure 2. Blank frames are not taken into account for the linguistic results. We consider redundant or the transitory stage between meaningful frames.

The loss for the encoder part is:

$$L_{ctc} = -\log p_{ctc}(\mathbf{y}|\mathbf{x}) \tag{4}$$

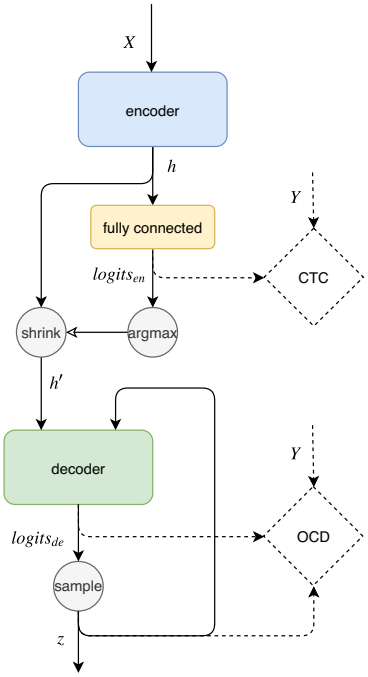

Fig. 1: *The framework of the* ECTC-DOCD *structure. The training and inferring share the same working flow demonstrated by the solid lines. The dash lines indicate the operations to compute the loss in the training phase.*

3.2. Decoder with OCD Loss

Similar to RNA, the decoder concatenates the embedded vector of the previously decoded label and current acoustic vector as the input frame for each step. As mentioned above, the decoder takes as input the shrunk hidden representation $\mathbf{h}'$ and would directly output the linguistic sequence:

$$p(z_u|h'_u, z_{u-1}) \triangleq \text{decoder}([h'_u; \text{embed}(z_{u-1})]) \tag{5}$$

During the training, however, the encoder could not guarantee the length of shrunk hidden representation equaling to the target sequence, even at the end of the training. Due to this, vanilla CE loss can not directly apply to the posterior distribution sequence $p_{\text{de}}(z)$ and the target sequence $\mathbf{y}$. Following the work [13], we apply Optimal Completion Distillation (OCD) to compute the CE loss (called *OCD loss*).

OCD takes the partly decoded result $z_{1:u}$ into consider and identifies the set of optimal suffixes that minimize the final edit distance by dynamic programming. The target distribution for each position of the generated sequence is constructed by putting equal probability to the tokens belong to the candidates of the optimal suffixes. OCD calculates the CE loss of the model's output with the optimal distribution at each time:

$$L_{ocd}(\mathbf{z}|\mathbf{h}') = \sum_{u=1}^{U} \text{KL}(p(z_u|h'_u, z_{u-1})||\text{OCD}(\mathbf{y}, \mathbf{z}_{1:u})) \tag{6}$$

Moreover, the usage of OCD makes the ECTC-DOCD avoid two mismatches: 1) the prefixes seen by the model during training and inference; 2) the training loss and the task evaluation metric. These mismatches are criticized [19, 20, 7, 21, 22]

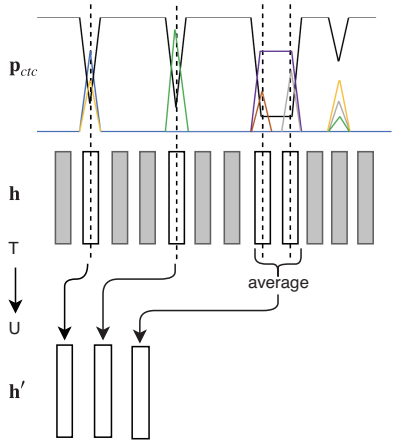

Fig. 2: *The shrinking layer to remove the blank frames (gray rectangles) in the acoustic representation. If there are repeated frames (white rectangles where no blank frames in between), an average operation will be applied to these frames and to obtain a single frame. The length of T shrinks into U through the shrinking layer.*

but commonly exist in Seq2seq models trained with Maximum Likelihood Estimation (MLE).

3.3. Systematic Learning

We propose a new learning paradigm called *systematic learning* for ECTC-DOCD. We interpret the supervision of CTC as representing the acoustic input and OCD as spelling the words. This design idea follows a previous work [6], making ECTC-DOCD more interpretable among the end-to-end models. During the training phase, the model needs to pay variational efforts on learning the two abilities, listen and spell, over time: the model focuses on learning a good representation for acoustic information in the beginning and gradually switches its efforts to spell what is heard. Following this opinion, systematic learning adds two losses as a final one with a *scheduled increasing* trade-off parameter λ ranging from 0 to 1:

$$L = (1-\lambda)L_{ctc}(\mathbf{y}|\mathbf{x}) + \lambda L_{ocd}(\mathbf{z}|\mathbf{h}') \tag{7}$$

Note that two supervisions share the same transcription, as in [1]. We can also use additional transcriptions, which use acoustic units like phones or syllables, for the encoder's supervision.

4. Experiments

We mainly conduct our experiments on the Chinese Mandarin speech AISHELL-2 [23], which is by far the largest free speech corpus available for Mandarin ASR research and contains 1000 hours for training. We extract 80-dimensional log-Melfilterbank features, operated with a 25ms window and shifted every 10ms. We compute the delta, delta-delta of the feature as the other two feature maps.

We first conduct several experiments to verify the rationality of the working flows designed in Section 3 for ECTC-DOCD. Then we infer the model with the first and second pass decoding [24] on several open source datasets, including LibriSpeech [25] and HKUST [26]. We compare our results with other baselines and state-of-the-art methods. Finally, we try to localize the limitation of ECTC-DOCD for further improvements.

4.1. Working Flow Explorations

Table 1 : *Some explorations of the working flows for the* ECTC-DOCD *and comparison on the Aishell2 dev. As a baseline, CTC model is* ECTC-DOCD*'s encoder. The criteria is CER(%).*

Working Flows	Models Settings	dev
CTC	baseline	12.75
ECTC-DOCD	hidden as input + average shrink + $\lambda = 0.2 \rightarrow 0.8$	**12.00**
decoder loss	MLE	12.50 (4.2% ↑)
acoustic input	distribution	12.64 (5.3% ↑)
	bottleneck=100	12.56 (4.3% ↑)
shrinking settings	weighted sum	12.13 (0.7% ↑)
	add adjacent frames	14.25 (18.3% ↑)
systematic learning	$\lambda = 0.8$	12.16 (1.3% ↑)
	$\lambda = 0.5$	12.04 (0.3% ↑)
	$\lambda = 0.2$	12.15 (1.2% ↑)
	$\lambda = 0.01 \rightarrow 0.99$	12.15 (1.2% ↑)

The base working flow of ECTC-DOCD follows the description in Section 3. We adopt the encoder designed in [27] for its promising results in Chinese ASR task, which roughly contains two CNN layers and four LSTM cells. CNN layers as the front-end part of several LSTM layers in the encoder is widely used in recent works [2, 1, 28]. The encoder downsamples the raw feature sequence by a rate of 8 in the time dimension and rate of 2 in the feature dimension. The decoder is a single LSTM cell with 800 hidden units. Confidence penalty [29] for the logits scaled with 0.3 is added to each loss.

To investigate how much improvement ECTC-DOCD will achieve from the decoder, we set a CTC baseline with the same encoder and training settings. For all the models in Tabel 1, only greedy search without language model is applied and LSTM cells are unidirectional.

4.1.1. Decoder Loss

As mentioned in Section 3.2, We recommend OCD training for the decoder because it can elegantly compute target(s) for each step according to the ground-truth. To validate its effectiveness, we alternatively use MLE without OCD. Contrary to the normal MLE under teacher-forcing training, we apply it under the self-dependent training [13]. It is actually scheduled sampling with always sampling [19].

To apply MLE, we assume the output of the model is consistent with target labels from left to right and ignore the mismatch part (if any) at the end of the two sequences. As shown in Table 1, OCD training can significantly outperform MLE.

4.1.2. Acoustic Input

Here we investigate the reasonability to use hidden representation $\mathbf{h}$ as the acoustic input. First, we make a comparison between $\mathbf{h}$ and the distribution p_{ctc}. Second, a smaller hidden size of acoustic representation $\mathbf{h}$ may force the encoder to learn a more compressed and general representation. We adopt the idea from transfer learning [30], the bottleneck, and additionally map the original representation to a narrower one before the final fully connected layer in the encoder.

As we can see from Table 1, using distribution as acoustic input for the decoder achieves worse performance. On the other hand, the performance is hindered rather than improved after introducing the bottleneck. We consider that distribution and bottleneck lose more information than the hidden representation as acoustic input.

4.1.3. Shrinking Settings

Next, we verify the effectiveness of simply averaging over the repeated frames to process the acoustic input. For comparison, we unequally treat these frames through the model's confidence in them. Specifically, a weighted sum over the repeated frames by the corresponding probabilities output p_{ctc} is computed. Additionally, considering CNN models usually achieve better performance with a wider horizon, we also let ECTC-DOCD see broader features for one step decoding. To realize this, we concatenate two adjacent frames with current one.

The results in Table 1 indicate that neither the two changes on shrinking setting can imrpove the results. We speculate that the clearity of the acoutic information is more important than capacity.

4.1.4. Systematic Learning

We investigate different schedules for λ in Eq 7. We simply apply a linear increase strategy as in [19] with two different settings: 1) starting to linearly increase λ at 10000-th step from 0.01 to 0.99 at 20000-th step during training. The baseline applies schedule: $0.2 \rightarrow 0.8$. For comparison, we apply three fixed settings ($\lambda = 0.2, 0.5, 0.8$).

According to Table 1, systematic learning with λ from 0.2 to 0.8 achieves the best performance.

4.2. First and Second Pass Decode

To compete with the state-of-the-art models, we perform first and second decoding by incorporating a neural language model [10] in this section. It is unnecessary to consider the end of decoding since the decoding steps are constrained by the length of shrunk acoustic representation $\mathbf{h}'$. We use self-attention structure as the language model [31] and trained with training text.

Besides, we use BLSTM with 800 hidden states for each direction to replace the LSTM in the encoder for fair comparison with other published results.

Table 2: *Performances (WER%) on test sets of Aishell2, Librispeech and HKUST.*

Model	Aishell2	Libri	HKUST
CTC	9.7	7.1	28.6
ECTC-DOCD	9.6	6.5	28.3
+ 1,2-pass	**8.5**	6.2	28.3
Chain-TDNN [23]	8.81	-	-
TDNN-hybrid+MMI [32]	19.78	**4.28**	28.2
DeepSpeech2 [2]	-	5.83	-
Self-Attention Aligner [4]	-	-	**25.88**

Table 2 compares our model with state-of-the-art models on three different datasets. As the end-to-end model, ECTC-DOCD can achieve competing results without the help of global attention mechanism. Nevertheless, on some datasets, ECTC-DOCD cannot significantly surpass our strong CTC baseline with the additional modelling ability for output labels. In Section 4.3, we emphasize on figuring out the bottleneck of the performance of ECTC-DOCD.

Table 3: *Performance (CER%) comparison before and after using flawless acoustic representation in* ECTC-DOCD *during inferring.*

Model	Description	train_dev	dev
Baseline	character transcriptions for both encoder and decoder	33.83	34.72
Model-A	holdout tran_dev transcriptions from training	34.25	36.48
Model-B	use tran_dev syllable transcriptions for training	**6.95**	36.47

4.3. Limitation of acoustic representation

In this section, we investigate on the limitation of ECTC-DOCD to improve the performance for further work. As demonstrated in Figure 1, the decoder of ECTC-DOCD can only peek the speech signal through the representations generated by the encoder, which are largely shaped by the CTC supervision. While in traditional end-to-end models, the hidden represention produced from the encoder are totally decided by the final supervision. We speculate that the performance of the whole model is restrained by the validity of the acoustic representation $\mathbf{h}$ or $\mathbf{h}'$.

To figure out how much potential improvement the model will gain with a better encoder, we plan to feed flawless acoustic representation to the decoder during testing. To ensure the encoder only delivers acoustic information rather the target labels to the decoder, syllable as units [33] is instead used to supervise the encoder, as explained in Section 3.3.

We construct experiments on HKUST dataset. The size of characters for the decoder is 3673 and syllables for the encoder is 1385. We first train the ECTC-DOCD, holding out train_dev sets in the standard way, to get the first model (Model-A). Then we obtain a second model (Model-B) by further training the first model's encoder with train_dev's syllable transcriptions. As shown in Table 3, it is noticeable that the second model's decoder has a dramatic improvement on the train_dev set.

We consider that the unnoticed differences among the various working flows in Section 4.1 are attributed to the limitation of the acoustic representation, which is mainly learned by the CTC supervision. In further works, it is worth exploring how to make the decoder robust to the fed acoustic representation or additionally utilise the original speech features.

5. Conclusions

This work presents ECTC-DOCD, an innovative end-to-end structure for the ASR task. ECTC-DOCD contains two level supervisions: CTC loss in the encoder and OCD loss in the decoder, which are respectively used for acoustic representation and linguistic generation. The shrinking layer used in ECTC-DOCD provides a bran-new method to bridge acoustic and linguistic parts in end-to-end models. ECTC-DOCD's training phase takes the task evaluation metric into account and exactly matches the inferring phase. Our model achieves promising results in Mandarin and English speech datasets. Further investigation will focus on how to make the decoder robust to the acoustic representation from the encoder.

6. References

[1] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4835–4839.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[3] A. Zeyer, K. Irie, R. Schlüter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," *arXiv preprint arXiv:1805.03294*, 2018.

[4] L. Dong, F. Wang, and B. Xu, "Self-attention aligner: A latency-control end-to-end model for asr using self-attention network and chunk-hopping," *arXiv preprint arXiv:1902.06450*, 2019.

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

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[7] 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.

[8] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Proc. Interspeech*, 2017, pp. 1298–1302.

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

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[13] S. Sabour, W. Chan, and M. Norouzi, "Optimal completion distillation for sequence learning," *arXiv preprint arXiv:1810.01398*, 2018.

[14] 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*. ACM, 2006, pp. 369–376.

[15] Z. Chen, Y. Zhuang, Y. Qian, K. Yu, Z. Chen, Y. Zhuang, Y. Qian, K. Yu, K. Yu, Y. Zhuang *et al.*, "Phone synchronous speech recognition with ctc lattices," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 1, pp. 90–101, 2017.

[16] S. Toshniwal, 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.

[17] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[19] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[20] D. Bahdanau, D. Serdyuk, P. Brakel, N. R. Ke, J. Chorowski, A. Courville, and Y. Bengio, "Task loss estimation for sequence prediction," *arXiv preprint arXiv:1511.06456*, 2015.

[21] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[22] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, "An actor-critic algorithm for sequence prediction," *arXiv preprint arXiv:1607.07086*, 2016.

[23] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.

[24] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.

[25] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[26] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, "Hkust/mts: A very large scale mandarin telephone speech corpus," in *Chinese Spoken Language Processing*. Springer, 2006, pp. 724–735.

[27] L. Dong, S. Zhou, W. Chen, and B. Xu, "Extending recurrent neural aligner for streaming end-to-end speech recognition in mandarin," *arXiv preprint arXiv:1806.06342*, 2018.

[28] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[29] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," *CoRR*, vol. abs/1701.06548, 2017. [Online]. Available: http://arxiv.org/abs/1701.06548

[30] W. Pan, E. Zhong, and Q. Yang, "Transfer learning for text mining," in *Mining Text Data*. Springer, 2012, pp. 223–257.

[31] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, "Character-level language modeling with deeper self-attention," *arXiv preprint arXiv:1808.04444*, 2018.

[32] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *Interspeech*, 2016, pp. 2751–2755.

[33] 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.