INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Weakly Supervised Construction of ASR Systems from Massive Video Data

Mengli Cheng[#], Chengyu Wang[#], Jun Huang, Xiaobo Wang

Alibaba Group, Hangzhou, China

{mengli.cml,chengyu.wcy,huangjun.hj,yongshu.wxb}@alibaba-inc.com

Abstract

Despite the rapid development of deep learning models, for real-world applications, building large-scale Automatic Speech Recognition (ASR) systems from scratch is still significantly challenging, mostly due to the time-consuming and financially-expensive process of annotating a large amount of audio data with transcripts. Although several self-supervised pre-training models have been proposed to learn speech representations, applying such models directly might be sub-optimal if more labeled, training data could be obtained without a large cost.

In this paper, we present VideoASR, a weakly supervised framework for constructing ASR systems from massive video data. As user-generated videos often contain human-speech audio roughly aligned with subtitles, we consider videos as an important knowledge source, and propose an effective approach to extract high-quality audio aligned with transcripts from videos based on text detection and Optical Character Recognition. The underlying ASR models can be fine-tuned to fit any domain-specific target training datasets after weakly supervised pre-training on automatically generated datasets. Extensive experiments show that VideoASR can easily produce state-of-the-art results on six public datasets for Mandarin speech recognition. In addition, the VideoASR framework has been deployed on the cloud to support various industrial-scale applications.

Index Terms: automatic speech recognition, weakly supervised learning, optical character recognition, massive video data

1. Introduction

Automatic Speech Recognition (ASR) is one of the core tasks in speech processing. The task aims to generate transcripts from speech utterances. Recently, end-to-end neural ASR models have been extensively studied, as these models do not require the explicit learning of acoustic and language models [1, 2].

Despite the success, a potential drawback is that these models require large amounts of transcribed data to produce satisfactory results [3]. Unfortunately, transcribing audio by human annotators is both time-consuming and financially-expensive [4]. Recently, self-supervised pre-training has been applied to ASR [5, 6], using unlabeled audio to pre-train ASR models. However, there exists a learning gap between pre-training objectives (such as minimizing the contrast loss [6]) and ASR training objectives. We assume that better results could be obtained if the models could optimize similar goals during pre-training and fine-tuning. Some methods generate synthetic speeches aligned with texts for training [7, 8], but the generated speeches may still be different from real ones.

In order to support real-world applications, a natural question arises: *is it possible to build accurate end-to-end ASR systems without much manually labeled data?* In this work, we present a weakly supervised framework to construct ASR systems from massive video data, named VideoASR. [1] The framework is shown in Figure 1, which consists of two major stages: *Weakly Supervised Pre-training* (WSP) and *Domain-specific Fine-tuning* (DF). During WSP, based on text detection [9] and Optical Character Recognition (OCR) [10], we extract human-speech audio aligned with subtitles from user-generated videos as knowledge sources to pre-train ASR models. Here, we pre-train our models over massive videos of varied topics so that the models can capture transferable, general knowledge across domains. The underlying ASR models can be fine-tuned to fit training data (usually smaller in size) in any domains. We evaluate VideoASR over popular ASR models and six public datasets. Results show that it produces state-of-the-art results for Mandarin speech recognition. VideoASR has also been deployed in an industrial-scale distributed machine learning platform to support various applications on the cloud.

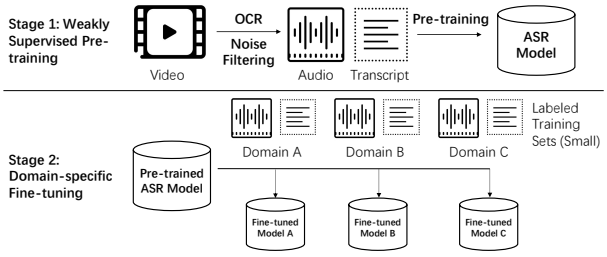


Figure 1: *High-level architecture of the VideoASR framework.*

2. Related Work

End-to-end ASR Models. While hybrid ASR techniques are continuously developing (such as classical DNN-HMM-style models [11]), due to the simple model pipelines, end-to-end ASR models have gained much attention. Recurrent-style networks are naturally suitable for end-to-end ASR as they model the sequences of audio and languages [12, 1, 13]; however, they may be slow during training and inference. This reduces the application scopes of such models in industry. CNN-based approaches [14, 15] are faster in speed, but they have limited capacity for modeling long sequences. Transformer-based methods [16, 17, 18, 19] have better performance because have strong abilities to capture long-term dependencies. They also converge faster and produce more accurate results when the CTC (Connectionist Temporal Classification) loss is added as an auxiliary loss [20]. Because the architecture design is not our major focus, we do not further elaborate.

Pre-training ASR Models. Two streams of works have been proposed to reduce the requirements of manually labeled data for end-to-end ASR. One stream applies unsupervised/self-

[#] Equal contribution.

[1] We categorize our framework to be *weakly supervised* because we use additional labeled data that are not processed by human annotators.

http://dx.doi.org/10.21437/Interspeech.2021-7

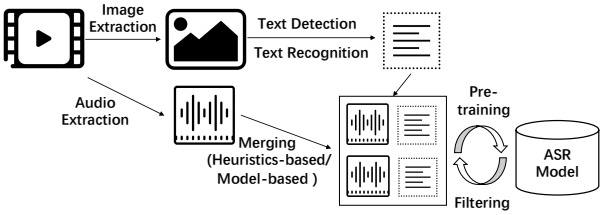


Figure 2: *Pipeline of Weakly Supervised Pre-training (WSP).*

supervised/semi-supervised methods to tackle the problem. For example, Long et al. [21] propose semi-supervised training of DNN and RNN based acoustic models. Chung et al. [22] introduce the Auto-regressive Predictive Coding for learning speech representations. The wav2vec framework [5, 6] introduces self-supervised pre-training for transformer-based ASR. Inspired by BERT [23], Baevski et al. [6] propose masked predictive coding for unsupervised pre-training of transformer encoders. Similar works include [24, 25]. Besides, a few methods aim to generate pseudo labels of unlabeled speeches for ASR training [26, 27, 28]. The other stream extracts aligned text-speech segments using existing ASR models. Lanchantin et al. [29] align paragraphs of transcripts with audio to generate training data. The works [30, 31] introduce several heuristic rules to extract useful speech segments with transcripts from Youtube. Different from previous methods, our work considers vision techniques to generate high-quality utterance-text pairs.

3. VideoASR: The Proposed Framework

In this section, we introduce technical details of our framework and the ASR model architectures that we use.

3.1. Weakly Supervised Pre-training

The pipeline of WSP is illustrated in Figure 2.

Video Acquisition. Many user-generated videos have embedded subtitles that are almost synchronous with the audio. We regard such videos as pre-training knowledge sources. The videos that we use are of various genres provided from *Youku* [2].

Text and Audio Spotting. Although videos with subtitles are available to us, subtitles are generally embedded in frame images in different styles and formats. This prevents us from extracting subtitles from raw data sources directly. Hence, we first extract frame images from each video with an interval of 1/3 second. Next, we employ IncepText [9] to detect text positions from images and the OCR model [10] to recognize texts.

Given a sequence of frame images within a time window size (denoted as $s_i, s_{i+1}, \cdots, s_{j-1}, s_j$), we wish to determine whether two consecutive frames s_k and s_{k+1} ($i \leq k, k+1 \leq j$) can be "merged" so that a subset of such frames may correspond to the same subtitle. Subsequently, the audio within the time frames is treated as the speech that roughly matches the subtitle. We present two merging methods: *Heuristics-based* and *Model-based*. For two consecutive frames s_k and s_{k+1}, denote the detected texts as t_k and t_{k+1}, respectively. Define the *Relative Edit Distance* (RED) between s_k and s_{k+1} as:

$$RED(s_k, s_{k+1}) = \frac{EditDis(t_k, t_{k+1})}{\max(Len(t_k), Len(t_{k+1}))}$$

where $EditDis(t_k, t_{k+1})$ is the *edit distance* between t_k and t_{k+1}, and $Len(t_k)$ is the length of t_k. *Heuristics-based Merging* combines two frames s_k and s_{k+1} if $RED(s_k, s_{k+1})$ is smaller than a tuned threshold.

However, *Heuristics-based Merging* may ignore the corresponding relations between audio and texts. If any existing ASR model is available, we can use it to refine the merging process [3]. Let a_k be the audio segment w.r.t. the frame s_k. *Model-based Merging* employs an existing model f to predict the transcript of a_k, denoted as $f(a_k)$. If s_k and s_{k+1} should not be merged, the error rate of model f is computed as:

$$Err_1(f, s_k, s_{k+1}) = CER(t_k, f(a_k)) + CER(t_{k+1}, f(a_{k+1}))$$

where $CER(t_k, f(a_k))$ is the Character Error Rate (CER) of model f's predictions. If s_k and s_{k+1} should be merged, similarly, we have the combined error rate:

$$Err_2(f, s_k, s_{k+1}) = \min\{CER(t_k, f(a_{k:k+1})), CER(t_{k+1}, f(a_{k:k+1}))\}$$

where $a_{k:k+1}$ concatenates a_k and a_{k+1}. s_k and s_{k+1} should be merged if $Err_1(f, s_k, s_{k+1}) > Err_2(f, s_k, s_{k+1})$.

Iterative Pre-training. After extraction and merging, we obtain a large "pseudo-labeled" dataset $D = \{(a_k, t_k)\}$, consisting of audio-transcript segment pairs. We pre-train the ASR model using the way as normal training over the dataset D. It should be noted that the extraction process of D unavoidably injects noise into the dataset due to the lack of human annotation. During pre-training, we apply a self-training strategy to filter out noisy data. In each epoch, we filter out audio-transcript segments $\{(a_k, t_k)\}$ from D that are most likely to have noisy transcripts and use the remaining dataset for the next training epoch. Due to space limitation, we omit the details and refer interested readers to [32].

3.2. Domain-specific Fine-tuning

Based on the WSP learning objective, our framework could generate ready-to-use ASR models directly. However, the domains of pre-training data may be significantly different from downstream ASR tasks. Hence, given a (small) training set $D_m = \{(a_k, t_k)\}$ of domain m, we fine-tune the pre-trained model over D_m to learn domain-adaptive parameters. One can also leverage transfer learning using both D and D_m to improve the fine-tuning performance, which is left as future work.

3.3. Choices of Model Architectures

Following industry practices, we consider two popular ASR models: wav2letter [14] and Speech Transformer [16], as shown in Figure 3. Wav2letter uses one dimensional convolution networks with large kernels as encoders, and the CTC loss for training. Its efficient inference speed makes it appealing to industrial applications. Speech Transformer [16] adopts self-attention for acoustic modelling and decoding. Following [20], the CTC loss is added as an auxiliary loss to achieve faster convergence and better performance. In multi-head attention layers, we set the hidden size as 512, with 8 heads. For fast online inference, we apply a beam search of size 16 for both models.

[2] *Youku* (`http://www.youku.com`) is a popular video hosting service, a subsidiary of Alibaba Group. It holds the copyrights of these videos, and permits authors to obtain and process the data as described. Different from standard movies, the user-generated videos usually do not have subtitles provided as standalone files.

[3] We use a commercial Madarin ASR service to transcribe the audios. See `https://ai.aliyun.com/nls/asr`.

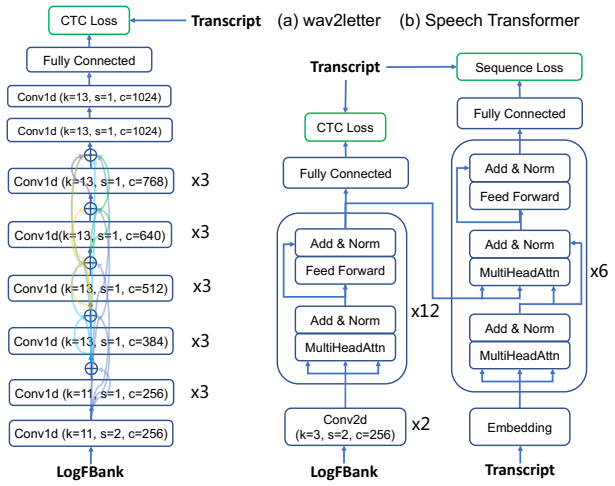


Figure 3: *The architectures of two models that we choose.*

Table 1: *Dataset statistics. (SPK: #Speakers. TXT: #Transcripts. UTT: #Utterances. R/S: Reading/spontaneous style.)*

Dataset	Duration	SPK	TXT	UTT	Style
ST_CMDS	500h	855	74,770	82,080	R
AISHELL-1	178h	400	113,738	120,099	R
AISHELL-2	1,000h	1,991	603,738	1,009,223	R
AIDATANG	200h	600	133,684	164,905	R
MagicData	760h	1,080	275,778	573,480	R
HKUST	200h	2,100	173,028	173,028	S

4. Experiments

In this section, we conduct extensive experiments to evaluate the proposed framework in various aspects. We also present the results of industrial deployment to demonstrate the effectiveness of the VideoASR framework.

4.1. Datasets and Experimental Settings

For ASR pre-training, we obtain 43,694 video clips of various topics from *Youku*. The total duration is around 8,000 hours. During WSP, the learning rates of wav2letter and Speech Transformer are set as 0.05 and 1.0, respectively. During fine-tuning, we set the learning rates to be 0.01 and 0.5. For both models, we normalize the utterances to 16kHz and generate the logarithm of FBank features of 80 dimensions, with a window size of 20ms and the stride of 10ms. SpecAugment [33] is applied for data augmentation. All algorithms are implemented in Tensorflow and run on GPU servers.

For evaluation, we follow the work [34] to evaluate the models on two popular Mandarin ASR datasets: AISHELL-1[4] and HKUST[5]. The baselines models include the classical TDNN model [13] [6] and the Speech Transformer with and without pre-training over unlabeled human speeches [16]. Specifically, in [16], over 10,000h unlabeled audios are employed to pre-train the model. We also consider four other public datasets, namely, ST_CMDS[7], AISHELL-2[8], AIDATANG[9] and MagicData[10]. The statistics of all six datasets are shown in Table 1. The datasets are varied in domains and styles and have relatively short duration compared to our WSP dataset. We keep the training, development and testing splits of all the datasets as default. In the experiments, we benchmark the two models (wav2letter and Speech Transformer) under two settings (with and without WSP) on all six datasets by ourselves.

4.2. Experimental Results and Model Analysis

General Performance. All the experimental results are summarized in Table 2. We have the following findings: i) Speech Transformer outperforms wav2letter across all the datasets. ii) The WSP technique in VideoASR effectively boosts the performance of both models on all the datasets. This phenomenon is more significant on small datasets (i.e., AIDATANG and HKUST). iii) Speech Transformer with WSP achieves state-of-the-art performance on all the six public datasets, outperforming baseline ASR approaches.

Analysis of WSP. We test both merging techniques via a manual check on 2,000 consecutive frame pairs that require proper merging. It shows model-based merging consistently produces better results (with a CER of 12.5%, compared to 33.0% for rule-based merging). This shows, even without human annotation, we can generate pre-training datasets with tolerable error rates. After text and audio spotting, we obtain a total of 1,825,927 utterances from all videos. The duration ranges from 15s to 20s. Next, we evaluate the iterative pre-training technique. We filter out part of the data (quantified by the drop ratio γ) and take the rest as the pre-training data for the next iteration. We search for the best value of γ from $0, 0.5\%, 1.0\%, 2.0\%$ and also compare our method with a classical data filtering approach [30]. We use the Mandarin ASR model from *https://ai.aliyun.com/nls/asr* for the implementation of [30], instead of their original English ASR model. In Table 3, we display the CER values produced by pre-trained wav2letter without fine-tuning, evaluated on the AISHELL-1 development set. It shows that WSP with $\gamma = 1.0\%$ has the best performance. After iterative pre-training, we obtain a "cleaner" pre-training ASR dataset. The CER of extracted speech-transcript pairs after noise filtering is around 6%, close to those of manually labeled datasets. [11]

Convergence analysis. During the DF stage, we investigate how WSP affects the DF performance. The convergence curves on HKUST are shown in Figure 5. As seen, wav2letter and Speech Transformer converge within 10 and 3 training epochs, respectively. Compared to the same models without WSP, the speed of convergence is much faster for both models, which clearly indicates WSP is able to find better parameter initialization for domain-specific ASR tasks, no matter whether there exist domain differences between the two datasets.

Error analysis and case studies. We analyze the percentages of different error types occurred in the test sets of AISHELL-1 and HKUST, shown in Table 4. The underlying ASR models are Speech Transformer w. and w/o. WSP. The majority of the errors are substitution errors caused by homophones. The WSP

[4] http://www.aishelltech.com/kysjcp/

[5] https://catalog.ldc.upenn.edu/LDC2005S15/

[6] Benchmark results of TDNN are from https://github.com/kaldi-asr/kaldi/blob/master/egs/aishell/s5/RESULTS and https://github.com/kaldi-asr/kaldi/blob/master/egs/hkust/s5/RESULTS.

[7] http://www.openslr.org/38/

[8] http://www.aishelltech.com/aishell_2/

[9] http://www.openslr.org/62/

[10] http://www.openslr.org/68/

[11] Our figure is computed over 0.2% of the pre-training dataset after noise filtering. The CERs of human-labeled data in AISHELL-1 and MagicData are close to 5% and 2%, respectively.

Table 2: *Performance of ASR models on public test datasets in terms of CER (%). * refers to our own implementation.*

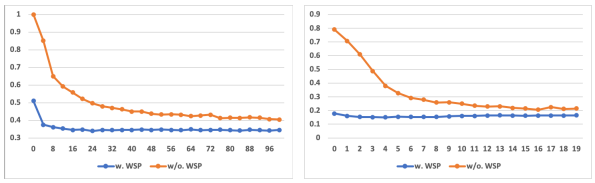
Model	ST_CMDS	AISHELL-1	AISHELL-2	AIDATANG	MagicData	HKUST
TDNN [13]	-	8.7	-	-	-	32.7
Transformer (w/o. pre-training) [34]	-	9.5	-	-	-	23.8
Transformer (w. pre-training) [34]	-	7.4	-	-	-	21.0
wav2letter w/o. WSP*	4.5	11.7	12.5	12.9	7.4	35.7
wav2letter w. WSP*	2.4	7.1	10.0	9.2	6.7	29.3
Transformer w/o. WSP*	4.4	6.7	7.4	7.8	3.6	23.5
Transformer w. WSP*	**2.1**	**5.9**	**5.9**	**4.9**	**3.3**	**20.0**

	Case A	Case B
Truth	送上**真挚**祝福 (Send **sincere** blessings)	今晚的比赛中朱婷**独得**27分 (Zhu Ting **alone scored** 27 points in tonight's game)
Output (w/o. WSP)	送上**真正**祝福 (Send **real** blessings) *Songshang **Zhenzheng** Zhufu*	今晚的比赛中朱婷**夺得**7分 (Zhu Ting **scored** 27 points in tonight's game) *Jinwan de Bisaizhong Zhuting **Duode** Ershiqifen*
Output (w. WSP)	送上**真挚**祝福 (Send **sincere** blessings) *Songshang **Zhenzhi** Zhufu*	今晚的比赛中朱婷**独得**27分 (Zhu Ting **alone scored** 27 points in tonight's game) *Jinwan de Bisaizhong Zhuting **Dude** Ershiqifen*

Figure 4: *Cases of model prediction w. and w/o. WSP. Italic texts refer to pronunciation (spelled in Mandarin phonetic symbols).*

Table 3: *Performance of pre-trained wav2letter with different data filtering techniques in terms of CER (%).*

Method/Iteration	4	8	12
Liao et al. [30]	17.3	16.8	16.5
WSP ($\gamma = 0$)	16.1	15.0	14.2
WSP ($\gamma = 0.5\%$)	15.4	14.4	13.6
WSP ($\gamma = 1.0\%$)	15.3	14.2	**13.3**
WSP ($\gamma = 2.0\%$)	15.6	14.9	14.7



(a) Model: wav2letter (b) Model: Speech Transformer

Figure 5: *Convergence curves on HKUST. (X-axis: number of epochs; Y-axis: CER on the development set.)*

technique helps to reduce such errors, as pronunciations and language contexts in the pre-training dataset are more diverse, leading to the better generalization ability of ASR models. Two typical cases are in Figure 4, with Chinese pronunciation and English translation provided. It shows WSP's ability to distinguish words with similar pronunciation.

4.3. Industrial Deployment

As our VideoASR framework is mostly designed for industrial-scale applications, we describe how the framework can be deployed on the cloud and further present the ASR results on our in-house dataset in real-world applications.

Platform deployment. The VideoASR framework has been deployed on the EasyASR distributed machine learning platform [35], which supports efficient training and serving large-scale ASR models across multiple workers and GPUs. Users also have the options to customize the end-to-end ASR model structures easily. Hence, our solution is highly applicable for industrial-scale scenarios.

Table 4: *Error analysis in terms of CER (%).*

Dataset	WSP?	Insertion	Deletion	Substitution
AISHELL-1	No	0.1	0.2	6.4
AISHELL-1	Yes	0.1	0.2	5.7
HKUST	No	2.6	3.6	17.3
HKUST	Yes	2.7	2.6	14.7

Table 5: *Performance of ASR models on our in-house dataset in terms of CER (%) and batch inference time.*

Model	CER	Inference Time
wav2letter w/o. WSP	25.3	2.68s
wav2letter w. WSP	**10.8**	2.66s
Speech Transformer w/o. WSP	16.8	15.57s
Speech Transformer w. WSP	**8.4**	15.68s

Evaluation on the in-house, e-commerce dataset. To further demonstrate the practical values of our work, we present the evaluation results on our in-house corpus. Our dataset is used in the e-commerce live-streaming domain, consisting of 350h human-speech audios with manually labeled transcripts. We hold out 2,761 audio clips (each around 15s) for evaluating the performance of wav2letter and Speech Transformer. Parameter settings are the same as in previous experiments. In the experiments, we report the performance on both model accuracy and batch inference time. Specifically, the inference time is measured using one Tesla V100 GPU (16GB), with the batch size set to be 32. The experimental results are summarized in Table 5. We can see that our pre-training technique significantly improves the model accuracy in both cases. In addition, although wav2letter is relatively inaccurate, its fast inference speed makes it appealing to real-time applications.

5. Conclusion and Future Work

In this paper, we present VideoASR and construct accurate ASR systems based on the weak supervision of massive video data. With WSP and the Speech Transformer model with our modifications, we achieve the state-of-the-art results on several public datasets. We also deploy our framework in the distributed machine learning platform and achieve desirable performance on our in-house datasets. Future work includes i) applying our approach to other languages and ASR models; ii) combining unsupervised and weakly supervised pre-training for ASR; and iii) leveraging transfer learning to improve model fine-tuning.

6. References

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

[2] G. Saon, Z. Tüske, K. Audhkhasi, and B. Kingsbury, "Sequence noise injected training for end-to-end speech recognition," in *ICASSP*, 2019, pp. 6261–6265.

[3] J. Hsu, Y. Chen, and H. Lee, "Meta learning for end-to-end low-resource speech recognition," in *ICASSP*, 2020.

[4] C. Manolache, A. Georgescu, A. Caranica, and H. Cucu, "Automatic annotation of speech corpora using approximate transcripts," in *TSP*, 2020, pp. 386–391.

[5] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019, pp. 3465–3469.

[6] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[7] Y. Chen, Z. Yang, C. Yeh, M. Jain, and M. L. Seltzer, "Aipnet: Generative adversarial pre-training of accent-invariant networks for end-to-end speech recognition," in *ICASSP*, 2020, pp. 6979–6983.

[8] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *ICASSP*, 2020.

[9] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "Inceptext: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection," in *IJCAI*, 2018, pp. 1071–1077.

[10] C. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *CVPR*, 2016, pp. 2231–2239.

[11] T. Tanaka, R. Masumura, T. Moriya, T. Oba, and Y. Aono, "A joint end-to-end and DNN-HMM hybrid automatic speech recognition system with transferring sharable knowledge," in *Interspeech*, 2019.

[12] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech*, 2015, pp. 3214–3218.

[13] 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.

[14] R. Collobert, C. Puhrsch, and G. Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[15] Z. Huang, T. Ng, L. Liu, H. Mason, X. Zhuang, and D. Liu, "SND-CNN: self-normalizing deep cnns with scaled exponential linear units for speech recognition," in *ICASSP*, 2020, pp. 6854–6858.

[16] Y. Zhao, J. Li, X. Wang, and Y. Li, "The speechtransformer for large-scale mandarin chinese speech recognition," in *ICASSP*, 2019, pp. 7095–7099.

[17] N. Moritz, T. Hori, and J. L. Roux, "Streaming automatic speech recognition with the transformer model," in *ICASSP*, 2020, pp. 6074–6078.

[18] K. J. Han, J. Huang, Y. Tang, X. He, and B. Zhou, "Multi-stride self-attention for speech recognition," in *Interspeech*, 2019, pp. 2788–2792.

[19] X. Chang, W. Zhang, Y. Qian, J. L. Roux, and S. Watanabe, "End-to-end multi-speaker speech recognition with transformer," in *ICASSP*, 2020, pp. 6134–6138.

[20] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, "Transformer-based online ctc/attention end-to-end speech recognition architecture," in *ICASSP*, 2020.

[21] Y. Long, Y. Li, S. Wei, Q. Zhang, and C. Yang, "Large-scale semi-supervised training in deep learning acoustic model for ASR," *IEEE Access*, vol. 7, 2019.

[22] Y. Chung, W. Hsu, H. Tang, and J. R. Glass, "An unsupervised autoregressive model for speech representation learning," in *Interspeech*, 2019, pp. 146–150.

[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

[24] A. T. Liu, S. Yang, P. Chi, P. Hsu, and H. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP*, 2020, pp. 6419–6423.

[25] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP*, 2020, pp. 6889–6893.

[26] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," in *Interspeech*, 2020, pp. 1006–1010.

[27] D. S. Park, Y. Zhang, Y. Jia, W. Han, C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," in *Interspeech*, 2020, pp. 2817–2821.

[28] K. Singh, V. Manohar, A. Xiao, S. Edunov, R. B. Girshick, V. Liptchinsky, C. Fuegen, Y. Saraf, G. Zweig, and A. Mohamed, "Large scale weakly and semi-supervised learning for low-resource video ASR," in *Interspeech*, 2020, pp. 3770–3774.

[29] P. Lanchantin, M. J. F. Gales, P. Karanasou, X. Liu, Y. Qian, L. Wang, P. C. Woodland, and C. Zhang, "Selection of multi-genre broadcast data for the training of automatic speech recognition systems," in *Interspeech*, 2016, pp. 3057–3061.

[30] H. Liao, E. McDermott, and A. W. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription," in *ASRU*, 2013, pp. 368–373.

[31] E. Lakomkin, S. Magg, C. Weber, and S. Wermter, "Kt-speech-crawler: Automatic dataset construction for speech recognition from youtube videos," in *EMNLP*, 2018, pp. 90–95.

[32] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2u-net: A simple noisy label detection approach for deep neural networks," in *ICCV*, 2019, pp. 3325–3333.

[33] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019, pp. 2613–2617.

[34] D. Jiang, X. Lei, W. Li, N. Luo, Y. Hu, W. Zou, and X. Li, "Improving transformer-based speech recognition using unsupervised pre-training," *arXiv preprint arXiv:1910.09932*, 2019.

[35] C. Wang, M. Cheng, X. Hu, and J. Huang, "Easyasr: A distributed machine learning platform for end-to-end automatic speech recognition," in *AAAI (Demo Track)*, 2021.