A Further Study of Unsupervised Pre-training for Transformer Based Speech Recognition

Dongwei Jiang, Wubo Li, Ruixiong Zhang, Miao Cao, Ne Luo, Yang Han, Wei Zou, Xiangang Li

AI Labs, Didi Chuxing, Beijing, China

{jiangdongwei, liwubo, zhangruixiong, caomiao, luone, hanyang, zouwei, lixiangang}@didiglobal.com

Abstract

Building a good speech recognition system usually requires large amounts of transcribed data, which is expensive to collect. To tackle this problem, many unsupervised pre-training methods have been proposed. Among these methods, Masked Predictive Coding achieved significant improvements on various speech recognition datasets with BERT-like Masked Reconstruction loss and Transformer backbone. However, many aspects of MPC have not been fully investigated. In this paper, we conduct a further study on MPC and focus on three important aspects: the effect of pre-training data speaking style, its extension on streaming model, and how to better transfer learned knowledge from pre-training stage to downstream tasks. Experiments reveled that pre-training data with matching speaking style is more useful on downstream recognition tasks. A unified training objective with APC and MPC provided 8.46% relative error reduction on streaming model trained on HKUST. Also, the combination of target data adaption and layer-wise discriminative training helped the knowledge transfer of MPC, which achieved 3.99% relative error reduction on AISHELL over a strong baseline.

Index Terms: unsupervised pre-training, transformer, speaking style, streaming speech recognition, knowledge transfer

1. Introduction

Current industrial end-to-end automatic speech recognition (ASR) systems rely heavily on large amounts of high quality transcribed audio data. However, transcribed data take substantial effort to obtain in industrial applications, while at the same time, a lot of un-transcribed data exist in online systems and cost little to collect. It is worthwhile to explore how to effectively use un-transcribed data to improve the performance of speech recognition systems when labeled data are limited.

Recently, unsupervised pre-training has shown promising results in several areas, including Computer Vision (CV) [1], Natural Language Processing (NLP) [2] and so on. One work that stands out among these methods is Bidirectional Encoder Representations from Transformers (BERT) [2], which used a Masked Language Model (MLM) pre-training objective and obtained new state-of-the-art results on eleven NLP benchmarks.

In speech area, researchers also proposed many unsupervised pre-training algorithms. Contrastive Predictive Coding (CPC) [3] extracts representation from data using contrastive loss that requires distinguishing a true future audio sample from negatives. There are multiple concurrent approaches that generalize this approach [4, 5] and applied it in learning speaker representation [6], extracting speech representation [7, 8], performing various speech-related tasks like speech recognition [7, 9, 10], speech emotion recognition [11] and so on. [12, 13, 14] proposed Autoregressive Predictive Coding

(APC) objective that predicts unseen future frame given past frames and also achieved good results on phonetic classification, speech recognition, and speech translation. Some other work [15, 16, 17, 18] got motivation from NLP and applied similar methods on speech tasks.

Among these methods, Masked Predictive Coding (MPC) [18] achieved significant improvements on state-of-the-arts Transformer based speech recognition models on various datasets without introducing any additional parameters for the speech recognition model. However, many aspects of MPC have not been fully explored:

- Speaking style has a strong impact on performance of ASR systems [19, 20]. Whether speaking style of pre-training data would affect the performance of downstream tasks has not been fully investigated.

- The ability to perform streaming recognition is important for ASR systems. MPC may not work for streaming recognition because it requires bidirectional context for predictive coding. How MPC can contribute to streaming models is worth exploring.

- Despite abundant work [21, 22, 23] on NLP about knowledge transfer between pre-trained model and downstream task, there are few work exploring how to perform better knowledge transfer in the area of speech.

In this paper, we investigate these aspects of MPC and discuss how we can extend MPC for better speech recognition.

2. Masked Predictive Coding

Masked Predictive Coding (MPC) [18] uses Masked Reconstruction objective to perform predictive coding on Transformer based models. As depicted in Fig. 1, the model structure of MPC is essentially the encoder part of Transformer based speech recognition model plus a single fully-connected projection layer. During training, masks are applied on input FBANK features before feeding into the encoder. The training objective is L1 loss computed between masked input FBANK features and projected encoder output at corresponding position. The local smoothness of speech makes the task of predicting adjacent frames too easy. So, in the current setup of MPC, we divide input features into chunks of four frames and apply mask on chunks with a probability of 15%. Unlike BERT, MPC adopted dynamic masking proposed in [24] where the masking pattern is generated every time a sequence is fed to the model.

One unique characteristic of sequence-to-sequence with attention ASR models is it usually applies downsampling in the encoder. Previous research showed temporal pooling encourages effective encoding in different temporal resolution and makes alignments in the decoding easier [25]. When temporal downsampling is applied, with input feature $X \in \mathbb{R}^{t \times d}$ and
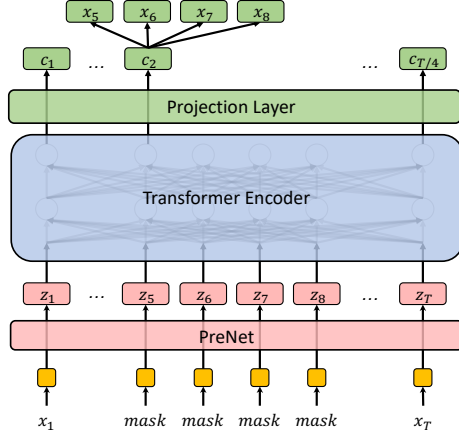
Figure 1: *Masked Predictive Coding with four-fold downsample*

temporal downsampling rate r, Transformer encoder projects the output of last encoder layer to dimension $X_0 \in \mathbb{R}^{t/r, d \times r}$. We then reshapes it back to same shape as input feature for MPC loss computation.

The overall training procedure of MPC consists of two stages, pre-training on unsupervised data and fine-tuning on supervised data. In the pre-training stage, MPC performs predictive coding directly on FBANK input and encoder output. After the unsupervised pre-training procedure, we remove the additional projection layer for predictive coding and add Transformer decoder for fine-tuning on downstream ASR tasks. All model parameters are end-to-end trainable in the fine-tuning stage. The work perhaps most similar to ours is Mockingjay [15], which also employed Transformer encoder with Masked Reconstruction loss. But their work mainly used pre-trained model as a feature extractor while MPC works more like BERT and focuses on obtaining a good parameter initialization. Also, unlike [15] and other previous work on predictive coding [7, 12], our setup does not introduce any additional parameters into speech recognition model.

3. Methods

3.1. MPC for streaming models

To apply MPC in streaming models, the Transformer encoder needs to be restricted to only use information that has appeared before. Though some previous work [26, 27] employed chunk-wise splitting for streaming models, in this paper, we simply changed self-attention mask on Transformer encoder to make the whole model stream-able. Specifically, we use a triangular matrix for self-attention mask $\mathbf{M}$ in encoder, where the upper triangular part is set to $-\infty$, and the other elements to 0.

Recent work on APC [13] got impressive results on downstream tasks with Transformer decoder backbone. Inspired by [28], we also propose to use a unified training objective that combines MPC and APC. During training, with probability **p**, we apply triangular matrix on Transformer encoder and use APC objective, with probability **1 - p**, we use the Transformer encoder as-is with MPC objective. This parameter sharing framework has the advantage of making the learned speech representations more general because they are jointly optimized for different pre-training objectives where context is utilized in different ways.

3.2. Knowledge transfer for MPC

For speech recognition task in a specific domain, its data distribution may be a lot different from data used for MPC pre-training. Directly using MPC model in fine-tuning stage might cause degradation in performance, even catastrophic forgetting [29, 30]. To deal with this problem, we followed previous work [21, 23] and adopted **target data adaption** to fine-tune MPC model on data of target task before the fine-tuning stage.

It is well-known that different layers of neural network capture different types of information, and the transferability of different layers also varies a lot [31, 32]. Previous work [21] made use of these findings by assigning lower learning rates to layers that are more general for downstream tasks and achieved promising results. To adapt their findings to MPC, we first used probing task [33, 34] on pre-trained model to find out which layers of Transformer encoder are more useful downstream speech recognition tasks. After that, **layer-wise discriminative training** [21] is used to assign different learning rates to each layer of Transformer encoder and adapt them to different extents for better knowledge transfer.

Transfer Learning with single-step auxiliary loss [22] is yet another transfer learning approach that adopted multi-task learning perspective via the addition of pre-training objective in the fine-tuning stage. With the same idea, we also added MPC loss in the fine-tuning stage. This way, the joint loss in the fine-tuning stage became the weighted sum of task-specific loss L_{Attn}, L_{CTC} and auxiliary MPC loss L_{MPC}:

$$L = \alpha_{attn} * L_{Attn} + \beta_{ctc} * L_{CTC} + \gamma_{mpc} * L_{MPC}, \quad (1)$$

For convenience, we name this method **multi-task with MPC** through out this paper. Masked input and output for MPC are also added during training.

4. Experiments

4.1. Data

For Mandarin pre-training, we first collected reading type dataset OpenMandarin as described in [18]. Note unlike [18], we did not include HKUST and AISHELL in OpenMandarin dataset in this work because we found self pre-trained MPC also improves performance. OpenMandarin contains about 1150 hours of speech in total. To understand the impact of pre-training data speaking style, our internal reading type dataset Didi Dictation and spontaneous type dataset Didi Callcenter were also included. Didi Dictation is collected from our internal mobile dictation application while Didi Callcenter is collected from phone calls between our user and customer service staff. We randomly selected 5000 hours of Didi Callcenter and 5000 hours of Didi Dictation for MPC pre-training. For fair comparison with Open Mandarin, we also added another pre-training dataset by randomly selecting 1150 hours of Didi Callcenter and name it Didi Callcenter - 1K. For English pre-training data, reading type dataset Librispeech [35] and spontaneous type dataset Fisher [36] are used. To provide a fair comparison on data size, we created a new pre-training dataset by randomly selecting only 960 hours of Fisher dataset and name it Fisher - 1K. Detailed information of pre-training corpora is provided in Table 1.

Fine-tuning experiments were conducted on HKUST, AISHELL-1 and Switchboard. The speaking style of HKUST and Switchboard is spontaneous while the speaking style of AISHELL is reading. Speed perturbation of 0.9 and 1.1 was used on the training data. Mandarin characters are used as

Table 1: *Details of pre-training corpora*

Datasets	Hours	Speaking Style
Open Mandarin	1150	Reading
Didi Callcenter - 1K	1150	Spontaneous
Didi Callcenter	5000	Spontaneous
Didi Dictation	5000	Reading
Librispeech	960	Reading
Fisher - 1K	960	Spontaneous

modeling units for HKUST and AISHELL like described in [37], while 2000 BPE subwords [38] is used for experiments on Switchboard.

The sample rate of HKUST and Switchboard is 8000 Hz, which is lower than some pre-training data. To alleviate the possible influence, we kept the sample rate of target dataset unchanged and downsampled pre-training data with higher sample rate to 8000 Hz when needed.

4.2. Experimental setups

For Transformer based full-sequence models, we followed model structure of previous work [39] with $e = 12$, $d = 6$, $d_{model} = 256$, $d_{ff} = 1280$ and $d_{head} = 4$. A source sequence X is first fed into a prenet consisting of two-layer CNN with 256 channels, stride size 2 and kernel size 3 and transformed to subsampled sequence $X_0 \in \mathbb{R}^{n^{sub} \times d^{attn}}$ before feeding into Transformer. For Transformer based streaming models, two representative work with CTC [40] and transducer loss [41] are slightly modified and used in this work. We used the same prenet and encoder structure as Transformer based models. A two-layer Transformer decoder is used as prediction network for transducer.

In pre-training stage, all models are trained with a total batch size of 256 for about 100 epochs. We used Adam optimizer and varied learning rate with warmup schedule [42] according to the formula:

$$lrate = k * d_{model}^{0.5} * min(n^{-0.5}, n * warmup\_n^{-1.5}), \quad (2)$$

where n is the step number. $k = 0.5$ and $warmup\_n = 5000$ were chosen for all pre-training experiments.

In the fine-tuning stage, all models are trained with a total batch size of 128. The same warmup schedule is used except we changed n to 1.0 and $warmup\_n$ to 25000. For Transformer based models, the Attention-CTC multi-task training objective is used, with weights determined on development set. Models for HKUST and AISHELL is trained for 50 epochs while model for Switchboard is trained for 100 epochs. Label smoothing of 0.1 and weight decay of $1e^{-5}$ are applied for all fine-tuning experiments. SpecAugument [43] is applied when training for Switchboard.

In the decoding stage, we selected 10 models with lowest error rates on the validation set and averaged their parameters. Beam search with Attention-CTC joint decoding and RNN language model [44] is used for Transformer based models, with weights determined using grid search on development set. The beam size is 10 for HKUST and AISHELL for Transformer based models and 20 for Switchboard. WFST with word-level language model is used for decoding with CTC.

We made our code publicly available for reproducibility at `https://github.com/athena-team/athena/tree/mpc_improvement`.

Table 2: *Character Error Rates(%) and Relative Error Rates Reduction(%) on HKUST and AISHELL test set*

Task	Pre-training Data	Hours	CER	RERR
HKUST	-	-	23.3	-
	HKUST	170	22.8	2.14
	Open Mandarin (8k)	1150	22.7	2.58
	Didi Callcenter - 1K	1150	22.0	5.58
	Didi Callcenter	5000	**21.5**	**7.73**
	Didi Dictation (8k)	5000	22.1	5.15
AISHELL	-	-	6.82	-
	AISHELL	178	6.61	3.07
	Open Mandarin	1150	6.38	6.45
	Didi Dictation	5000	**6.26**	**8.21**

Table 3: *Word Error Rates(%) and Relative Error Rates Reduction(%) on Switchboard and CallHome test set*

Pre-training Data	Hours	WER		RERR	
		SWB	CH	SWB	CH
-	-	8.8	17.8	-	-
Switchboard	260	8.5	17.4	3.41	2.25
Fisher - 1K	960	**8.0**	**16.2**	**9.09**	**8.99**
Librispeech (8k)	960	8.4	17.2	4.55	3.37

4.3. Effect of pre-training data speaking style

The results on HKUST and AISHELL with different pre-training data are listed in Table 2. Our baseline result without MPC matches the strong baseline in [39]. Comparing relative error reduction of HKUST with same amounts of pre-training data, it is obvious MPC models pre-trained with matching speaking style data achieved lower error rates for downstream tasks.

Open Mandarin and Didi Dictation are both used in pre-training for HKUST and AISHELL. From Table 2, we can also find the relative error reduction using the same pre-training data is bigger with matching fine-tuning data. Interestingly, the error reduction Didi Callcenter - 1K brings on HKUST is even bigger than Didi Dictation, which suggests speaking style has a bigger impact than pre-training data size in some cases.

Experiments were also conducted on English ASR database Switchboard. Our baseline results on Switchboard is also on par with previous work [39]. As shown in Table 3, Switchboard got better results with spontaneous type pre-training data Fisher - 1K than with Librispeech, which further confirms our findings on speaking style.

4.4. MPC for streaming model

We first tested the effect of directly initializing streaming model with MPC. As shown in Table 4, pre-trained MPC brings consistent gains on streaming models for HKUST and AISHELL. MPC model pre-trained on more data obtained better results on streaming models, which is also in line with the conclusion for full-sequence models. Though not specifically optimized, MPC pre-training is also useful for streaming models, which suggests the parameter initialization obtained by MPC is helpful for both uni-directional and bidirectional models.

To further improve performance of MPC on streaming models, we tried to combine APC with MPC in pre-training

Table 4: *Character Error Rates(%) and Relative Error Rates Reduction(%) for uni-directional CTC and uni-directional RNN-T with pre-trained MPC*

Task	Model	Pre-training Data	CER	RERR
HKUST	CTC	-	29.3	-
		Didi Callcenter - 1K	28.0	4.56
		Didi Callcenter	**27.6**	**5.80**
	RNN-T	-	28.1	-
		Didi Callcenter - 1K	26.9	4.43
		Didi Callcenter	**26.6**	**5.20**
AISHELL	CTC	-	9.91	-
		Open Mandarin	**9.42**	**4.91**
	RNN-T	-	9.43	-
		Open Mandarin	**8.88**	**5.83**

Table 5: *MPC + APC is the model pre-trained with APC 50% of time and MPC 50% of time. Relative Error Rates Reduction(%) is calculated with HKUST baseline without MPC*

Pre-training Data	Objective	CER	RERR
Didi Callcenter - 1K	MPC	28.0	4.56
	APC	27.8	5.12
	MPC + APC	**27.2**	**7.32**
Didi Callcenter	MPC	27.6	5.80
	APC	27.9	4.72
	MPC + APC	**26.8**	**8.46**

stage. Previous work on APC showed a future prediction step of 5 gave best results on Transformer decoder [13]. We implemented APC with the same future prediction step and set switching probability **p** to 0.5. Experiments were conducted on HKUST dataset with Transformer CTC model and the results of different pre-training objectives are presented in Table 5. When used alone, APC and MPC got similar improvements. Combining APC with MPC results in consistent gains over them, which echoes with the findings in [28].

4.5. Knowledge transfer for MPC

Each layer of pre-trained MPC encoder captures different features of the input speech. To get a quantitative measure of how important layer *l* is, for each layer, we create a new Transformer with *l* layers of encoder and 6 layers of decoder. The encoder for the new Transformer is frozen and initialized from pre-trained MPC model while the decoder is trainable. As shown in Fig 2, features from the middle layers of pre-trained Transformer encoder are generally more helpful than features from top and bottom layers. In the fine-tuning stage, we propose to set the learning rates of encoder layers discriminatively by multiplying the learning rate of layer *l* with $\lambda^{\|l-\theta\|}$, where $\lambda \in (0, 1)$. To fit the accuracy curve above, we set λ to be 0.95 and θ to be 5.5. This way, the parameters of middle layers are updated more slowly and knowledge from pre-trained MPC model is best retained. Different λ and θ values have also been tested and no significant difference is observed. Using this schedule on baseline systems without MPC doesn't lead to better accuracy either.

The results for target data adaptation and layer-wise discriminative training are listed in Table 6. Two baseline re-
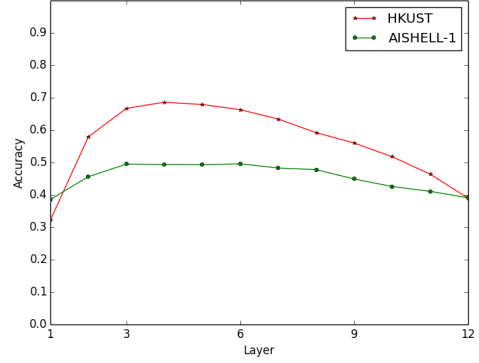

Figure 2: *CER(%) with each layer of pre-trained Transformer encoder. Results on HKUST is pre-trained with Didi Callcenter and results on AISHELL is pre-trained with Open Mandarin*

Table 6: *Results on HKUST and AISHELL with different knowledge transfer methods. HKUST is pre-trained with Didi Callcenter. AISHELL is pre-trained with Didi Dictation. Target Data + Layer-wise is the combination of target data adaption and layer-wise discriminative training*

Task	Knowledge Transfer Methods	CER	RERR
HKUST	-	21.5	-
	Target Data Adaption	21.2	1.40
	Layer-wise Discriminative	21.3	0.93
	Target Data + Layer-wise	**20.8**	**3.26**
	Multi-Task MPC	21.1	1.86
AISHELL	-	6.26	-
	Target Data Adaption	6.23	0.48
	Layer-wise Discriminative	6.21	0.80
	Target Data + Layer-wise	**6.01**	**3.99**
	Multi-Task MPC	6.13	2.08

sults we used are HKUST pre-trained with Didi Callcenter and AISHELL pre-trained with Open Mandarin. Using target data adaption or layer-wise discriminative training alone doesn't help the knowledge transfer of MPC very much. But when combined together, they provide consistent gains on downstream tasks.

For multi-task with MPC, the MPC loss should contribute more to the joint loss to facilitate knowledge transfer in the first few epochs. As training proceeds, the task-specific component of the loss function becomes more important and γ_{mpc} should be decreased. In this work, we empirically found it work well to set initial γ_{mpc} to 0.2 and decrease it by half every 5 epochs. The result in Table 6 shows multi-task with MPC gets slight improvements over baseline.

5. Conclusion

In this work, we investigated three important aspects of MPC. Pre-training data with matching speaking style was found to be more useful on downstream recognition tasks. Using MPC directly on streaming models helps, but combining MPC with APC brings further improvements on streaming models. Also, the combination of target data adaption and layer-wise discriminative training provides consistent gains on knowledge transfer to downstream tasks.

6. References

[1] C. Doersch, A. Gupta, and A. Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015, pp. 1422–1430.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pretraining of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*, 2019, pp. 4171–4186.

[3] O. A. van den, Y. Li, and V. Oriol, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[4] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[5] A. Baevski, M. Auli, and A. rahman Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," *arXiv preprint arXiv:1911.03912*, 2019.

[6] R. Mirco and B. Yoshua, "Learning speaker representations with mutual information," *Interspeech*, Sep 2019.

[7] S. Steffen, B. Alexei, C. Ronan, and A. Michael, "wav2vec: Unsupervised pre-training for speech recognition," *Interspeech*, Sep 2019.

[8] P. Santiago, R. Mirco, S. Joan, B. Antonio, and et al, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *Interspeech*, Sep 2019.

[9] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," *arXiv preprint arXiv:2001.11128*, 2020.

[10] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *arXiv preprint arXiv:2002.02848*, 2020.

[11] Z. Lian, Y. Li, J.Tao, and J. Huang, "Improving speech emotion recognition via transformer-based predictive coding through transfer learning," *arXiv preprint arXiv:1811.07691*, 2018.

[12] C. Yu-An, H. Wei-Ning, T. Hao, and G. James, "An unsupervised autoregressive model for speech representation learning," *Interspeech*, Sep 2019.

[13] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," *arXiv preprint arXiv:1910.12607*, 2019.

[14] C. Yu-An and G. James, "Improved speech representations with multi-target autoregressive predictive coding," *arXiv preprint arXiv:2004.05274*, 2020.

[15] A. T. Liu, S. Yang, P.-H. Chi, P.-C. Hsu, and et al, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," *arXiv preprint arXiv:1910.12638*, 2019.

[16] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," *arXiv preprint arXiv:2001.10603*, 2020.

[17] X. Song, G. Wang, Z. Wu, Y. Huang, and et al, "Speech-xlnet: Unsupervised acoustic model pretraining for self-attention networks," *arXiv preprint arXiv:1910.10387*, 2019.

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

[19] B. Mohamed, D. Renato, D. Olivier, D. Stephane, and et al, "Automatic speech recognition and speech variability: A review," *Speech communication*, vol. 49, no. 10-11, pp. 763–786, 2007.

[20] W. Mitch, T. Kelsey, H. Kate, and S. Amy, "Effect of speaking style on lvcsr performance," in *Proc. ICSLP*, vol. 96, 1996.

[21] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *ACL*, 2018.

[22] A. Chronopoulou, C. Baziotis, and A. Potamianos, "An embarrassingly simple approach for transfer learning from pretrained language models," *Proceedings of the 2019 Conference of the North*, 2019.

[23] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" *arXiv preprint arXiv:1905.05583*, 2019.

[24] Y. Liu, O. Myle, G. Naman, J. Du, and et al, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[25] 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.

[26] T. N. Sainath, C.-C. Chiu, R. Prabhavalkar, A. Kannan, and et al, "Improving the performance of online neural transducer models," in *ICASSP*, 2018, pp. 5864–5868.

[27] H. Miao, G. Cheng, C. Gao, P. Zhang, and Y. Yan, "Transformer-based online ctc/attention end-to-end speech recognition architecture," *arXiv preprint arXiv:2001.08290*, 2020.

[28] L. Dong, N. Yang, W. Wang, F. Wei, and et al, "Unified language model pre-training for natural language understanding and generation," in *NeurIPS*, 2019.

[29] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and et al, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[30] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, and et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, p. 3521–3526, Mar 2017.

[31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *NIPS*, 2014.

[32] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and et al, "Linguistic knowledge and transferability of contextual representations," *arXiv preprint arXiv:1903.08855*, 2019.

[33] X. Shi, I. Padhi, and K. Knight, "Does string-based neural mt learn source syntax?" in *EMNLP*, 2016.

[34] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, and et al, "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks," *arXiv preprint arXiv:1608.04207*, 2017.

[35] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.

[36] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text," in *LREC*, 2004.

[37] W. Zou, D. Jiang, S. Zhao, G. Yang, and et al, "Comparable study of modeling units for end-to-end mandarin speech recognition," in *ISCSLP*, 2018, pp. 369–373.

[38] T. Zenkel, R. Sanabria, F. Metze, and A. H. Waibel, "Subword and crossword units for ctc acoustic models," in *Interspeech*, 2018.

[39] K. Shigeki, C. Nanxin, H. Tomoki, H. Takaaki, and et al, "A comparative study on transformer vs rnn in speech applications," *arXiv preprint arXiv:1909.06317*, 2019.

[40] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *ICASSP*, 2019, pp. 7115–7119.

[41] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, and et al, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, and et al, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[43] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, and et al, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.

[44] K. Suyoun, H. Takaaki, and W. Shinji, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*, Mar 2017.