

Incremental Learning for End-to-End Automatic Speech Recognition

Li Fu^{1,†}, Xiaoxiao Li^{1,†}, Libo Zi^{1,†}

¹Jingdong Digits Technology Holding Co., Ltd. (JD Digits), Beijing, China

{fuli3, lixiaoxiao10, zilibo}@jd.com

Abstract

We propose an incremental learning for end-to-end Automatic Speech Recognition (ASR) to extend the model’s capacity on a new task while retaining the performance on existing ones. The proposed method is effective without accessing to the old dataset to address the issues of high training cost and old dataset unavailability. To achieve this, knowledge distillation is applied as a guidance to retain the recognition ability from the previous model, which is then combined with the new ASR task for model optimization. With an ASR model pre-trained on 12,000h Mandarin speech, we test our proposed method on 300h new scenario task and 1h new named entities task. Experiments show that our method yields 3.25% and 0.88% absolute Character Error Rate (CER) reduction on the new scenario, when compared with the pre-trained model and the full-data retraining baseline, respectively. It even yields a surprising 0.37% absolute CER reduction on the new scenario than the fine-tuning. For the new named entities task, our method significantly improves the accuracy compared with the pre-trained model, i.e. 16.95% absolute CER reduction. For both of the new task adaptations, the new models still maintain a same accuracy with the baseline on the old tasks.

Index Terms: automatic speech recognition, end-to-end, incremental learning, knowledge distillation

1. Introduction

End-to-end Deep Neural Network (DNN) has become an emerging trend in the area of Automatic Speech Recognition (ASR), which enables an easier model building and training than traditional pipeline [1, 2]. In many real applications, the end-to-end ASR model is always required to recognize speech for a new task while maintaining performance on existing ones. For example, an ASR model is well trained on the past dataset, while new specific named entities of the application need to be added into the model’s capability. Or for cold start of a new ASR scenario, the model is required to be adapted to a new domain based on a small new dataset while inheriting the recognition ability from the old ASR model. Since the new named entities or the new scenario might be out the scope of the past training dataset, the performance is probably to be poor when directly using the old model on the new task.

To solve this problem, a simple but cumbersome method is to retrain the ASR model with a mixture of the new dataset and the past dataset. However, the method might suffer from a data imbalance problem as the new dataset is usually much smaller than the past dataset [3, 4]. As new tasks continue to increase, repeated retraining on the cumbersome dataset becomes infeasible. In addition, the past dataset may be

unavailable due to data security, privacy, etc. [5]. Despite the shortcomings, the performance of retraining can be regarded as a baseline of our proposed method. An alternative method is to apply a specific-designed language model for a refined decoding or a post correction, but the essential problem arising from the acoustic model is still unsolved [6, 7].

Ideally, the new task could be learned with sharing parameters of the existing model, without degrading performance on the old tasks. Fine-tuning is usually applied to adjust to the new task by modifying the parameters of the pre-trained model [8]. Jiang et al. proposed an unsupervised method called masked predictive coding to obtain a pre-trained model for ASR fine-tuning [9]. Chuang et al. presented SpeechBERT for end-to-end spoken question answering tasks [10]. Although efficient for new task adaptation, fine-tuning dramatically degrades on the previous task since the shared parameters is modified with a stand-alone attention on the new task but lacking the guidance from the previous task-specific prediction.

Alternatively, incremental learning gains more attention to optimize models for new task without sacrificing model accuracy on old ones [11]. The existing incremental learning methods with DNN are mainly divided into three categories, i.e. without using old data, using synthetic data, and using exemplars from old data [12]. Although the latter two categories might improve the performance by introducing old data information, much workload on synthesis algorithm and sampling strategy is required. Also, they would not work if the dataset for previously learned tasks is unavailable. Without using old data, knowledge distillation could be applied to guide the new task learning, which provides promising potential on maintaining the performance on old tasks [13].

The aim of our work is to obtain a new end-to-end ASR model with incremental capability on the new task while without catastrophic forgetting, based on the old pre-trained model and the new dataset with a small size. First, we build a twin of DNNs, whose two network architectures are the same with the old pre-trained ASR model. Then, knowledge distillation [14] is applied to obtain the knowledge of the frozen old model in the twin networks. It is used as guidance for maintaining performance on the old dataset. Finally, the new model in the twin networks is fine-tuned to fit the new dataset with a combination of the distilling and the recognition task. To verify the performance of our proposed method, an end-to-end Connectionist Temporal Classification (CTC) ASR model based on self-attention structures is presented [15]. Experiments show that our proposed method performs excellent in new scenario task and new named entities task while maintaining previous knowledge.

To the best of our knowledge, our method is the first work that uses incremental learning for end-to-end ASR. The main contributions of our work are shown as follows.

[†] co-first authors

1. We present an incremental learning algorithm for end-to-end ASR, which can extend the model's capacity on a new task while maintaining the performance on existing ones.

2. Using the small new dataset, our method is efficient in training computation and simple in implement. It can also solve the situation when the old dataset is unavailable.

3. The accuracy of the proposed method outperforms the cumbersome retraining and even the fine-tuning on the new task. And it achieves an approximate accuracy on the old task, when compared with the retraining and the pre-training.

The remainder of this paper is organized as follows. The details of the proposed incremental learning for end-to-end ASR are presented in section 2. Section 3 is the experimental results. Finally, the conclusions are given in Section 4.

2. Incremental Learning for ASR

2.1. Problem Statement

Denote the training dataset for incremental learning ASR as $D_n = \{\mathbf{x}_n^i, \mathbf{y}_n^i | i \in [1, N_n]\}$, where $n \in \{1, 2\}$ indicates the new dataset and the old dataset, respectively. The dataset sample $\{\mathbf{x}_n^i, \mathbf{y}_n^i\}$ consists of a sequence audio feature vector \mathbf{x}_n^i with a unified dimensional, and the corresponding label sequence \mathbf{y}_n^i [16]. N_n is the number of samples in the training dataset. Usually the size of the new dataset is much smaller than the old one, i.e. $N_1 \ll N_2$.

Denote the end-to-end ASR model as a nonlinear function $f(\mathbf{y}^i | \mathbf{x}^i, \theta)$, where θ is the model parameters to be determined. Using the old dataset D_2 , the pre-trained model can be obtained as $f(\mathbf{y}^i | \mathbf{x}^i, \theta_2)$, which is trained to fit

$$\max_{\theta_2} \sum_{i=1}^{N_2} \log p(\mathbf{y}_2^i | \mathbf{x}_2^i, \theta_2) \quad (1)$$

The aim of the proposed incremental learning is to train a new end-to-end ASR model based on the new dataset D_1 and the previous pre-trained model $f(\mathbf{y}^i | \mathbf{x}^i, \theta_2)$ without accessing to the old dataset D_2 . The model is trained to fit the new task and maintain the performance on the old dataset, i.e.

$$\max_{\hat{\theta}_2} \sum_{i=1}^{N_1} \log p(\mathbf{y}_1^i | \mathbf{x}_1^i, \hat{\theta}_2) \quad (2)$$

$$\text{and } \sum_{i=1}^{N_2} \log p(\mathbf{y}_2^i | \mathbf{x}_2^i, \hat{\theta}_2) \approx \sum_{i=1}^{N_2} \log p(\mathbf{y}_2^i | \mathbf{x}_2^i, \theta_2) \quad (3)$$

where $\hat{\theta}_2 = \theta_2 + \Delta\theta_2$ is obtained by modifying the parameters of the pre-trained model θ_2 . A key property of the proposed incremental learning for end-to-end ASR is Equation (3), which means that the new model has an approximate performance on the old dataset, when compared with the old pre-trained model. Without Equation (3), the method is degenerated to fine-tuning. Although the fine-tuning may perform well in the new task, it suffers from performance degrading on the old dataset.

For the retraining baseline, the new dataset D_1 and the old dataset D_2 is mixed to be $D_0 = D_1 \cup D_2$ to obtain a model $f(\mathbf{y}^i | \mathbf{x}^i, \theta_0)$, which is trained to fit

$$\max_{\theta_0} \sum_{n=1}^2 \sum_{i=1}^{N_n} \log p(\mathbf{y}_n^i | \mathbf{x}_n^i, \theta_0) \quad (4)$$

2.2. Deep Incremental Learning

We apply a modified version of the existing CTC based end-to-end ASR model as an example to verify the performance of our approach. In reality, our approach can be easily extended to other end-to-end ASR frameworks, such as Recurrent Neural Network Transducer (RNN-T) [17], Listen Attend and Spell (LAS) [18]. Considering the advantages in sequence feature extraction and parallel-in-time computation of the transformer structures [19], we use Self-Attention Blocks (SABs) replacing the RNN cells of the existing CTC based ASR model as done in [20]. The model architecture is successively stacked with 3 Convolutional Neural Networks (CNNs), 10 SABs and 2 Fully Connected (FC) layers. The schematic representation of incremental learning for end-to-end ASR in this work is shown in Figure 1.

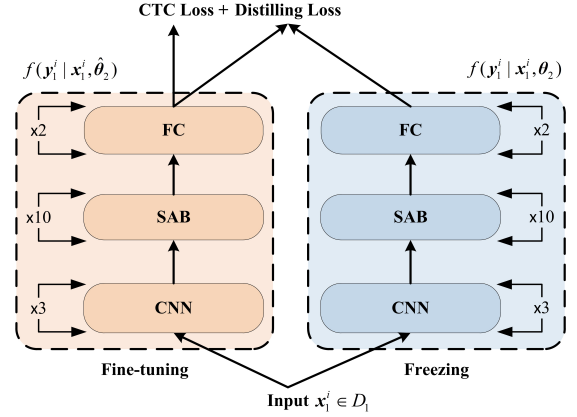


Figure 1: A schematic representation of incremental learning for end-to-end ASR.

Due to the missing of the old dataset in our method, we can only use the parameters of the pre-trained model to guide the training process retaining the recognition ability of the previous model. To achieve this, we build a twin of DNNs, whose two architectures are the same with the old end-to-end ASR model. And the parameters of the twin networks are initialized from the pre-trained network. One of the networks represents to the new model to be fine-tuned for the new task, and the other is frozen during training to obtain the knowledge of the old model, as shown in Figure 1. The new model is fine-tuned using new dataset with a combination of the CTC loss [21] and the distilling loss [14]. The CTC loss is used to train the model fitting to the new dataset, and the distilling loss is applied to guide the new model obtaining the same outputs with the old pre-trained model.

Given an input $\mathbf{x}_1^i \in D_1$, the sequence outputs of the new model and the old model of the twin networks are denote as $\Psi_n(\mathbf{x}_1^i)$, which consist of a time sequence output vectors $\pi_n^k(\mathbf{x}_1^i) \in \mathbf{R}^M$, where $k \in [1, K]$ and K is the length of the

output sequence, M is the number of modeling unit class plus a special symbol blank for ASR task.

We denote the CTC loss of the new model as [21]

$$L_{CTC} = \frac{1}{N_1} \sum_i^{N_1} l_{ctc}(\Psi_1(\mathbf{x}_1^i), \mathbf{y}_1^i) \quad (5)$$

The distilling loss of \mathbf{x}_1^i is calculated by Kullback-Leibler divergence, as

$$l_{KL}(\Psi_2(\mathbf{x}_1^i), \Psi_1(\mathbf{x}_1^i)) = \sum_{k=1}^K \sum_{m=1}^M p_{2,m}^k \log(p_{2,m}^k / p_{1,m}^k) \quad (6)$$

where $p_{n,m}^k = \exp(\pi_{n,m}^k(\mathbf{x}_1^i) / T) / \sum_{m=1}^M \exp(\pi_{n,m}^k(\mathbf{x}_1^i) / T)$, and T is the temperature scalar.

As the old model is frozen during new model training, the Equation (6) can be simplified to

$$l_{KL}(\Psi_2(\mathbf{x}_1^i), \Psi_1(\mathbf{x}_1^i)) = - \sum_{k=1}^K \sum_{m=1}^M p_{2,m}^k \log(p_{1,m}^k) \quad (7)$$

Since there are many elements with small value in the output of a converged CTC model [21], Equation (7) is modified to improve the numerical stability for incremental learning, we have

$$\begin{aligned} \log(p_{1,m}^k) &= \log \frac{\exp(\pi_{1,m}^k(\mathbf{x}_1^i) / T - s_1^k(\mathbf{x}_1^i) / T)}{\sum_{m=1}^M \exp(\pi_{1,m}^k(\mathbf{x}_1^i) / T - s_1^k(\mathbf{x}_1^i) / T)} \\ &= \pi_{1,m}^k(\mathbf{x}_1^i) / T - s_1^k(\mathbf{x}_1^i) / T - \log(S(\mathbf{x}_1^i, T)) \end{aligned} \quad (8)$$

where we have $S(\mathbf{x}_1^i, T) = \sum_{m=1}^M \exp(\pi_{1,m}^k(\mathbf{x}_1^i) / T - s_1^k(\mathbf{x}_1^i) / T)$, and $s_1^k(\mathbf{x}_1^i) = \max_m (\pi_{1,m}^k(\mathbf{x}_1^i))$, which ensures $S(\mathbf{x}_1^i, T) > 1$ for numerical stability.

Then the distilling loss for the new dataset is obtained as

$$L_d = \frac{1}{N_1} \sum_{i=1}^{N_1} l_{KL}(\Psi_2(\mathbf{x}_1^i), \Psi_1(\mathbf{x}_1^i)) \quad (9)$$

Finally, the overall loss for our proposed incremental learning for end-to-end ASR combines the CTC loss and the distilling loss as follows:

$$L = \lambda L_{CTC} + (1 - \lambda) \sigma L_d \quad (10)$$

where the parameter λ is used to balance between the two terms. If $\lambda=1$, the training process is degenerated to fine-tuning, which only pays attention on the new dataset and lacks the guidance from the previous task-specific prediction. If $\lambda=0$, the model will only retain the characteristics of the old pre-trained model without learning for the new task. The scalar σ is used to ensure that the two terms are in an order of magnitude. In our work, we fix $\lambda=0.5$ while change σ around 0.02 to test the effect of parameters on performance.

3. Results and Discussions

3.1. Dataset

The overall training dataset is 12,301 hours of Mandarin speech, which consists of 301 hours new dataset and 12000 hours old dataset. Two new datasets of JD Digits are separately collected as 300 hours for a new scenario task and 1 hour for a new named entities task, and the old dataset covers

internal speech dataset in the old business scopes of JD Digits and public speech dataset, AISHELL-1 [22], AISHELL-2 [23], THCHS-30 [24], Primewords [25], ST-CMDS [26], etc. To test the effectiveness of our method, Chinese character is used as the ASR modeling unit in our experiments [27]. In the dataset, the number of tokens on the modeling unit is 7228.

3.2. Experimental Setup

We use 80-dimension Mel-filter bank (fbank) features computed on 20ms window with 10ms shift [28]. The model architecture is composed of 3 CNNs, with filter dimension and channels (41,11,31), (21,11,32), (21,11,96) and the strides is (2,2), (2,1), (2,1) [15], which are followed by 10 SABs with 8 multi-head and 256 dimension. The output dimension of the penultimate FC and the last FC is 1024 and 7229, respectively.

The model is pre-trained based on the old dataset using the CTC loss, which is optimized in the same method as [20]. And we set warm up steps to 8000 and the parameter of learning rate to 0.5. All of our experiments are performed with a mini batch of 128 on 4 NVIDIA V100 GPUs. For Mandarin speech recognition, a 4-gram language model with pruning is trained on about 30GB cleaned text using the KenLM toolkit [29].

3.3. Compared Methods

Our proposed incremental learning for end-to-end ASR is compared with three commonly used methods, i.e. pre-training, retraining, and fine-tuning. As shown in Table 1, our method has a low training cost without accessing to the old dataset. Experiments show that our method performance excellently on the new task with few degrades on the old ones.

Table 1: Comparison between our method and others.

Methods	Pre-training	Retraining	Fine-tuning	Our Method
Dataset	D_2	$D_1 \cup D_2$	D_1	D_1
Data Size	12,000h	12,300h	300h/1h	300h/1h
Training Cost	High	High	Low	Low
Old Task	Good	Good	Bad	Good
New Task	Bad	Good	Good	Good

3.4. Results on New Scenario and New Named Entities

Our proposed incremental learning for end-to-end ASR with different parameters and the commonly used methods are evaluated on the old dataset and the new dataset, as summarized in Table 2 (new scenario) and Table 3 (new named entities). More in details, we use Character Error Rate (CER) for evaluation. The performance of the retrained model is regarded as a baseline of our proposed method.

Pre-training: As shown in Table 2, due to the missing of the new scenario dataset, the old pre-trained model achieves the worst performance on the new scenario, i.e. 2.37% absolute CER increment, when compared with the retraining baseline.

Fine-tuning: As shown in Table 2, although fine-tuning achieves -0.51% absolute CER reduction on the new scenario, it suffers from a dramatically degrading on the previous task since the shared parameters is modified with single attention on the new dataset but little guidance from the previous ability.

Our method: First, setting the temperature scalar $T=1$, we change the scalar σ in loss balance around 0.02 to test the

parameters effecting. As shown in Table 2, as σ decreases, the performance of our method on the new scenario improves, while the performance on the old test set gradually decreases. Experiments show that $\sigma=0.02$ performs the best on new task fitting and old task performance maintaining.

Then, we change the temperature scalar range from 1 to 5, and the results show robustness if $T > 1$. As referred in [14], $T > 1$ is suggested to increase the weight of smaller logits values and encourage the network to better encode similarities among classes. In our experiments, as T increases, the performance of our method on the new scenario improves slightly, while the performance on the old test set (more obvious on THCHS-30) slightly decreases. The performance on different test dataset with different temperature scalar is illustrated in Figure 2. Our results show that $T=3$ achieves the most excellent performance with a balance consideration of accuracy on the old dataset and the new dataset. It yields 0.88% and 3.25% absolute CER reduction, when compared with the retraining baseline and the pre-training, respectively. It even yields a surprise 0.37% absolute CER reduction, when compared with the fine-tuning. Moreover, the performance of our method on the old task achieves a very approximate performance with the retraining baseline in a small absolute CER difference from -0.18% to +0.37%.

Finally, unlike the new scenario with 300h training data, 1 hour of low-resource training dataset with new named entities is used to valid the performance of our approach. As shown in

Table 3, the pre-training achieves a very low accuracy on the new named entities. Although fine-tuning decreases 17.98% absolute CER on the new task, it suffers catastrophic forgetting. Similar with fine-tuning, our proposed method can also significantly improve the accuracy on the new named entities, i.e. 16.95% absolute CER reduction. Distinctively, our method achieves a very approximate performance with the pre-training baseline in a small absolute CER difference from +0.11% to +0.41%.

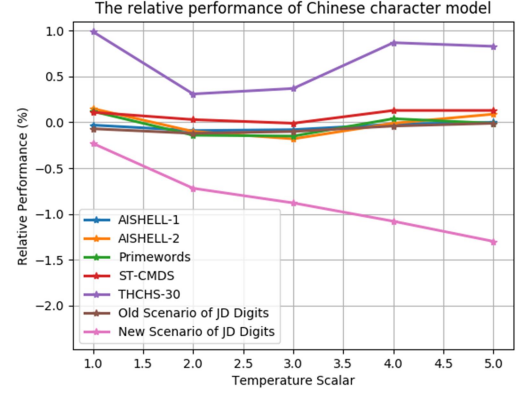


Figure 2: Performance on various test dataset with different temperature scalar.

Table 2: Results on the new scenario (values in bold font is the maximum difference with the retraining baseline).

CER/CER difference (%)		Old Task						New Task
		AISHELL-1	AISHELL-2	Primewords	ST-CMDS	THCHS-30	Old Scenario of JD Digits	New Scenario of JD Digits
Retraining (baseline)		1.13	4.98	4.10	2.26	11.48	5.53	7.70
Pre-training		-0.13	-0.39	-0.57	-0.18	-0.45	-0.11	+2.37
$T=1$	$\sigma=0$ (Fine-tuning)	+0.54	+1.10	+1.85	+1.33	+1.33	+0.60	-0.51
	$\sigma=0.02$	-0.03	+0.15	+0.12	+0.11	+0.99	-0.07	-0.23
	$\sigma=0.05$	-0.07	-0.01	-0.24	-0.05	+0.28	-0.07	+0.59
$\sigma=0.02$	$T=2$	-0.09	-0.10	-0.14	+0.03	+0.31	-0.12	-0.75
	$T=3$	-0.08	-0.18	-0.15	-0.01	+0.37	-0.10	-0.88
	$T=4$	-0.02	-0.01	+0.04	+0.13	+0.87	-0.04	-1.08
	$T=5$	+0.00	+0.00	-0.01	+0.13	+0.83	-0.01	-1.30

Table 3: Results on the new named entities (values in bold font is the maximum difference with the pre-training baseline).

CER/CER difference (%)		Old Task						New Task
		AISHELL-1	AISHELL-2	Primewords	ST-CMDS	THCHS-30	Old Scenario of JD Digits	New Named Entities
Pre-training (baseline)		1.00	4.59	3.53	2.08	11.03	5.42	18.47
Fine-tuning		+1.40	+2.49	+5.75	+2.51	+4.08	+0.35	-17.98
Our Method($\sigma=0.02, T=3$)		+0.11	+0.18	+0.41	+0.18	+0.38	+0.18	-16.95

4. Conclusions

The aim of incremental learning for end-to-end ASR is for the training model to adapt to the new speech recognition task without forgetting its existing knowledge. To achieve this, the new model is trained combining the task classification performance and the knowledge distillation of the old model without accessing to the old dataset. Besides low training

computation, our proposed method achieves the best performance with a significant absolute CER reduction on the new task when compared with other methods, while maintaining the performance on the old tasks. To valid the performance, we use 300h dataset and 1h dataset for the new scenario task and the new named entities task, respectively. In our future work, incremental learning for ASR with one shot will be investigated.

5. References

- [1] R. Prabhavalkar, K. Rao, T. Sainath, et al. “A comparison of sequence-to-sequence models for speech recognition,” in *INTERSPEECH – 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 939–943.
- [2] E. Battenberg, J. Chen, R. Child, et al., “Exploring neural transducers for end-to-end speech recognition,” *arXiv preprint arXiv: 1707.07413*, 2017.
- [3] J. Johnson and T. Khoshgoftaar, “Survey on deep learning with class imbalance,” *Journal of Big Data*, 2019, 6(1):1-54.
- [4] X. Qin, D. Cai, and M. Li, “Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation,” in *INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 4045–4049.
- [5] H. Bae, J. Jang, D. Jung, et al., “Security and privacy issues in deep learning,” *arXiv preprint arXiv: 1807.11655*, 2018.
- [6] T. Tanaka, R. Masumura, H. Masataki, Y. Aono, “Neural error corrective language models for automatic speech recognition,” in *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association*, 2018, pp. 401–405.
- [7] X. Liu, Y. Wang, X. Chen, et al., “Efficient lattice rescoring using recurrent neural network language models,” *IEEE ICASSP2014*. IEEE, 2014.
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [9] D. Jiang, X. Lei, W. Li, et al., “Improving transformer-based speech recognition using unsupervised pre-training,” *arXiv preprint arXiv: 1910.09932*, 2019.
- [10] Y. Chuang, C. Liu, and H. Lee, “SpeechBERT: cross-modal pre-trained language model for end-to-end spoken question answering,” *arXiv preprint arXiv: 1910.11559*, 2019.
- [11] F. Castro, M. Marin-Jimenez, N. Guil, et al., “End-to-end incremental learning,” In *The European Conference on Computer Vision (ECCV)*, 2018.
- [12] Y. Wu, Y. Chen, L. Wang, et al. “Large scale incremental learning,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [13] Z. Li and D. Hoiem, “Learning without forgetting,” In *European Conference on Computer Vision*, 2016, pp. 614–629.
- [14] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in *NIPS Workshop*, 2014.
- [15] S. Amodei, R. Ananthanarayanan, J. Anubhai, et al., “Deep speech 2: end-to-end speech recognition in English and Mandarin,” in *International Conference on Machine Learning*, 2016, pp. 173–182.
- [16] G. Saon and J. Chien, “Large-vocabulary continuous speech recognition systems: a look at some recent advances,” in *IEEE Signal Processing Magazine*, 2012, 29(6): 18–33.
- [17] A. Graves, “Sequence transduction with recurrent neural networks,” in *Proceedings of the 29th International Conference on Machine Learning*, 2012.
- [18] W. Chan, N. Jaitly, Q. Le, et al., “Listen, attend and spell: a neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016.
- [19] S. Karita, N. Soplin, S. Watanabe, et al., “Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration,” in *INTERSPEECH – 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 1408–1412.
- [20] J. Salazar, K. Kirchhoff, and Z. Huang, “Self-attention networks for connectionist temporal classification in speech recognition,” *arXiv preprint arXiv: 1901.10055*, 2019.
- [21] A. Graves, S. Fernandez, F. Gomez, et al., “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.
- [22] H. Bu, J. Du, X. Na, et al., “AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline,” in *20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, 2017.
- [23] J. Du, X. Na, X. Liu, et al., “AISHELL-2: transforming Mandarin ASR research into industrial scale,” *arXiv preprint arXiv: 1808.10583*, 2018.
- [24] D. Wang and X. Zhang, “THCHS-30: a free Chinese speech corpus,” *arXiv preprint arXiv: 1512.01882*, 2015.
- [25] Primewords Information Technology Co., Ltd., Primewords Chinese Corpus Set 1, 2018, <https://www.primewords.cn>.
- [26] ST-CMDS-20170001_1, Free ST Chinese Mandarin Corpus.
- [27] L. Fu, X. Li, and L. Zi, “Research on modeling units of transformer transducer for Mandarin speech recognition,” *arXiv preprint arXiv: 2004.13522*, 2020.
- [28] L. Narayana and S. Kopparapu, “Choice of mel filter bank in computing MFCC of a resampled speech,” *arXiv preprint arXiv: 1410.6903*, 2014.
- [29] K. Heafield, “KenLM: faster and smaller language model queries,” in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011 pp. 187–197.