Applying Connectionist Temporal Classification Objective Function to Chinese Mandarin Speech Recognition

Pengrui Wang, Jie Li, Bo Xu

Interactive Digital Media Technology Research Center
Institute of Automation, Chinese Academy of Sciences, Beijing, China, P.R.China
{wangpengrui2015, jie.li, xubo}@ia.ac.cn

Abstract

In automatic speech recognition (ASR), connectionist temporal classification (CTC) is regarded as a method to achieve end-to-end system. Actually, not only characters (Chars) but also context independent phonemes (CI-Phns) or context dependent phoneme (CD-Phns) can be used as output units of CTC-trained neural network. The contribution of this paper mainly lies in three aspects: First, we trained CTC models with three different units (Char, CI-Phn and CD-Phn) on Chinese Mandarin. The CTC-trained CD-Phn model might be first realized on Mandarin speech recognition (SR). Second, we optimize the training and decoding procedures, which benefit our CTC-trained models. Our Char model, a real end-to-end system, achieves a character error rate (CER) of 34.22% on HKUST corpus which surpasses the result (39.70%) reported by EESEN. Additionally, our CD-Phn model outperforms our hybrid model. Finally, we build a CTC-based online system using unidirectional Long Short-term Memory (UniLSTM) with row convolution (RC), which achieves comparable performance with our bidirectional LSTM (BiLSTM).

Index Terms: connectionist temporal classification, end-to-end, row convolution, automatic speech recognition.

1. Introduction

There are two mainly frameworks for deep learning based ASR. One is hybrid system based on deep neural network-hidden Markov model (DNN-HMM), the other is end-to-end system which uses deep neural network (DNN) model alone. The neural networks often include two basic types, feedforward neural network (FNN) and recurrent neural network (RNN), especially LSTM based RNN.

DNN-HMM hybrid system is a development of the traditional Gaussian mixture model-hidden Markov model (GMM-HMM) hybrid system, in which HMMs act to normalize the temporal variability of speech signal, whereas GMMs compute the emission probabilities of HMM states. In DNN-HMM model, DNN works as a substitute of GMMs for its excellent modeling abilities in acoustic models, e.g., FNN in [1], LSTM in [2, 3]. Note that DNN is trained to classify speech frames into HMMs' states, so its output distributions are not emission probabilities of HMM states, the state priors are needed to fix them according to Bayesian theory.

Training hybrid systems are rather complex, they have multiple stages and are expertise-intensive tasks, which need lots of human effort. Complexities are major in two aspects. First, acoustic modeling typically requires various resources such as dictionaries and phonetic questions, which demand significant human effort and often prove critical to overall performance, so it will be a great restriction to ASR systems if they are inadequate. Second, per-frame target states are required for training DNN. The alignment information is usually obtained from a trained GMM-HMM, so GMMs need to be designed and will go through CI phones, CD states. Although in [4, 5], researchers propose to flat-start DNNs to acquire GMM-free DNN training system, but the GMM-free approach is still complicate especially when targets are CD states.

By comparison, training end-to-end systems are simpler than training hybrid ones, they require less human effort and fewer stages. Recently, there are three basic methods to establish an end-to-end system: CTC-trained model [6], RNN Transducer [7, 8] and attention-based model [9, 10, 11]. None of them depends on pronunciation dictionary, phonetic questions, GMMs and HMMs. In other words, given a corpus, an ASR system will be established only rely on DNNs (most are deep RNNs). What's more, an end-to-end system is not only easy to deploy, but also overcomes some defects of hybrid systems. First, it avoids the inconsistency in hybrid systems where objective function used to train the networks is different from the true performance measure (sequence-level transcription accuracy). Second, it exploits RNN's potential of global sequence modeling. In a hybrid system, this potential is hindered by HMM. Third, it has the facility of integrating grammar information in training, such as RNN Transducer and attention-based model.

The shortcomings of end-to-end systems are that they are not easy to be trained and doesn't have overwhelming advantage in performance and training speed compared with hybrid systems. Recently, Some researchers used CTC-trained phoneme models to improve hybrid systems and achieved success [12, 13]. These inspire us to do this work.

In this paper, we realize CTC-trained phoneme models on Chinese Mandarin and explore optimizations on CTC-based training and decoding. We investigate the CTC-trained networks with different output units on a Mandarin SR task HKUST, they are Chars, CI-Phns and CD-Phns. A character in Chinese is a single Chinese character, so our CTC-trained Char model is a real end-to-end system. Experiments show that our CTC-trained CD-Phn model outperforms our cross-entropy-trained CD states baseline system, a relative error reduction of 2.4% is observed when networks are BiLSTMs and 9.8% when networks are UniLSTMs with RC (UniLSTM-RCs). Our CTC-trained Char system performs better than EESEN's [14] on HKUST corpus, which mainly due to our training and decoding skills. It's worth noting that EESEN, introduced in [15], is an open source end-to-end system on ASR. In addition,

The work was supported by 973 Program in China, grant No. 2013CB329302.

978-1-5090-4294-4/16/$31.00 ©2016 IEEE

our experiments show that UniLSTM-RC has a close performance to BiLSTM, so it is a suitable substitution of UniLSTM in online systems although with a few time delays.

The rest of this paper is organized as follows. Sections 2 describes models in training, Section 3 describes key points in decoding. Section 4 provides experimental results and concluding remarks are given in Section 5.

2. Structures in Training

We use UniLSTM and BiLSTM as basic networks, they have been introduced in many papers. In the following sections we only detail CTC objective function and UniLSTM-RC model.

2.1. CTC in Training

In ASR, CTC has the ability to learn the alignments between speech frames and their transcript label sequences. Given a speech $X = (x_1, ..., x_t, ..., x_T)$ and its transcript (e.g., phonemes or characters) $z = (z_1, ..., z_u, ..., z_U)$ $(T \geq U)$, the objective function of CTC-trained RNN model is:

$$O = -\sum_{(X,z)\in Set} ln\ Pr(z|X) \tag{1}$$

where $Pr(z|X)$ is the log-likelihood of transcript labels given speech frames and Set means training set.

To calculate $Pr(z|X)$ efficiently, a new label, blank label (represented with -), which expresses meaningless label or no emitting label is defined. Then, blank labels are added into transcript z to let every label in z has a blank label on both sides. After that, we get z's augmented label sequence $L_{aug} = (-, z_1, -, ..., z_u, -, ..., -z_U, -)$. Finally, we use a HMM-like model to calculate $Pr(z|X)$, the model is constructed according to L_{aug} and illustrated by Fig. 1.

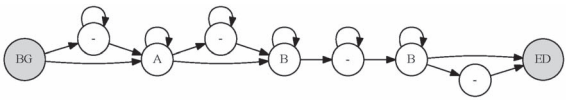

Figure 1: HMM-like model in CTC function when label sequence is $z = (A, B, B)$.

There are several points in the HMM-like model. Firstly, the "BG" state standing for entrance and "ED" state standing for exit are fake states, they just help to elaborate the model and can be removed. The other states each has only one emit symbol, which is drawn in its circle. Next, all transition paths are a fixed value of 1.0, which is different from standard HMM's paths. Finally, if two labels are same, they must go through state "-" just as the two "B" symbols in Fig. 1.

With the HMM-like model, $Pr(z|X)$ can be quickly calculated through a forward-backward algorithm as used in standard HMM. Once obtained $Pr(z|X)$, the objective function and its gradients is also calculated, and then backpropagation is executed. Details of calculations are in the seventh chapter of [16].

2.2. UniLSTM with Row Convolution

Experiments in [12, 13] show CTC using UniLSTM as the basic RNN (CTC-UniLSTM) is obviously inferior to CTC-BiLSTM. To improve the performance of CTC-UniLSTM, a RC layer is added upon the last hidden layer in DNN to provide the current frame with a few future information in [17].

We use τ to represent the number of future frames, so the input of RC layer is a matrix $H_{t:t+\tau} = [h_t, h_{t+1}, ..., h_{t+\tau}]$, h_t is the output from the last hidden layer at frame t. Assume the dimension of h_t is d, then the size of weight matrix W of RC layer is $(d, \tau+1)$. Then, the output at frame t can be calculated:

$$\tau_t = Diag(W H^T_{t:t+\tau}) \tag{2}$$

where T stands for matrix transpose and $Diag$ selects on-diagonal elements of a square matrix as a vector.

3. Decoding

3.1. Decoding with WFSTs

There are three methods for decoding CTC systems: best path decoding [6], prefix search decoding[6, 18] and WFST based decoding [15]. We adopt the last one because it's easy integrating grammar, lexicon and CD-Phns.

We need grammar WFST (denoted as G), spelling WFST (Ls), phoneme-lexicon WFST (Lp), CD-Phn to CI-Phn WFST (C), and three kinds of token WFSTs (T) for three output units respectively. Among them, G standing for language model, Lp standing for pronunciation dictionary and C standing for context dependence information have the same structure as they were used in hybrid system [19, 20], here Lp is denoted as L in [20]. Ls we used is illustrated in Fig. 2, where an optionally space character between every pair of words is to model word delimiting in the original transcripts. Token WFST maps a sequence of frame-level CTC labels to a single unit, it is similar to a function which merges continuously repeated labels and then deletes blank labels. For example, after processing several frames, neural network may generate 3 possible label sequences (only focus on part of labels and omit others) "... A A A A A A ...", "... - A - A A ...", "... A - - A ...". Token WFST of unit "A" maps all the shown labels in above 3 sequences into a singleton label "A", and then these sequences become like "... A ...". The character token WFST is shown in Fig. 3, the CI-Phn token WFST has the similar structure, but the CD-Phn token WFST is different. The generation of CD-Phns needs phoneme-tied, so a CD-Phn we used is physical CD-Phn, the structure of a physical CD-Phn is shown in Fig. 4.

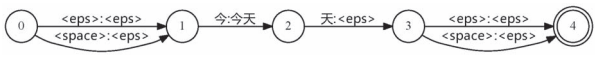

Figure 2: Spelling WFST for a Chinese word "Jin Tian". The "<eps>" symbol means no inputs are consumed or no outputs are emitted.

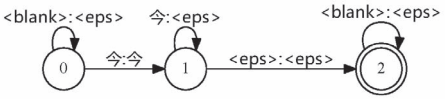

Figure 3: Character token WFST for a Chinese character "Jin". The "<blank>" symbol stands for blank label.

When decoding, we select needed WFSTs according to output unit type, and compose them to a search graph (denoted as S). If output units are Chars, S is composed by these FST operations:

$$S = T \circ \min\left(\det\left(Ls \circ G\right)\right) \tag{3}$$

where $\circ$, $\det$ and $\min$ denote composition, determinization and minimization respectively. When units are CI-Phns,

$$S = T \circ \min(\det(Lp \circ G)) \tag{4}$$

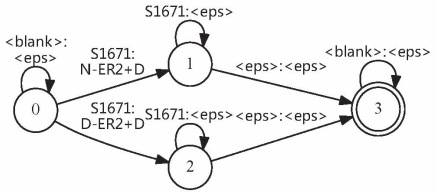

Figure 4: Token WFST of a physical CD-Phn "S1671". The symbols "N-ER2+D" and "D-ER2+D" are the logical CD-Phn correspond with "S1671".

S will be

$$S = T \circ \min(\det(C \circ (L \circ G))) \tag{5}$$

if units are CD-Phns.

3.2. Posterior Normalization in CTC

When decoding GMM-HMM models using WFST, the likelihoods from GMMs are used directly. However, in DNN-HMM models, the outputs of DNN are state posteriors, so state priors are needed when decoding. Similarly, CTC-trained models require label priors. In EESEN, the priors are computed from augmented label sequence. By analyzing the results in our experiments, we found that the insertion error is much higher than deletion error when using EESEN's method directly. This phenomenon might due to the high prior of the blank label. We add a cost to the blank label prior during decoding, which proves to be effective for performance gain.

4. Experiments

We do some explorations on CTC-based systems. These explorations mainly include learning rate adjustment strategy, blank label prior cost and UniLSTM-RC model for online system, they all give performance improvements.

4.1. Experiments Setup and Baseline

The corpus used in our experiments is HKUST [21], it's a large vocabulary Chinese Mandarin conversational telephone speech recognition task. The original corpus consists of 873 calls about 170-hours in training set and 24 calls about 5-hours in development set. We take its development set as testing set. Our development set is randomly selected from the training set, it has 22 calls. Then, most of data preparations are obtained from EESEN's hkust recipe "v1".

Deep RNNs we use are deep BiLSTM and deep UniLSTM. Each model has 3 hidden layers, each layer has 800 memory cells (or 400 cells in each direction of BiLSTM). We find that training LSTM with CTC objective function is easily divergent, so we limit the output of LSTM cells and the gradients of all parameters into range (-50, 50). What's more, we adopt the Max-Norm Regularization [22] and sort training set speeches in an ascending order by their frame counts.

The feature for each frame is 40-dimensional Mel-scale Log-FilterBank Coefficients (LFB), with its first order and second-order. We try using 80-dimensional LFB or concatenate frames with leaping frames as feature, just as [13] did, but neither of them brings performance gain.

Table 1 shows our baselines, they are all hybrid systems. LSTM models in baselines are similar to what we have described above, while FNN-HMM use 5 layers. Compared with [23], our baselines is trustworthy, we use fewer parameters counts but gain similar performances.

Table 1: Baselines about hybrid system

Models	TrnLAcc(%)	DevLAcc(%)	CER(%)
FNN-HMM	55.32	51.17	39.71
UniLSTM-HMM	66.24	61.36	34.34
BiLSTM-HMM	71.98	64.63	32.60

4.2. Learning Rate Adjustment Strategy

We define label error rate as LER, label accuracy as LAcc. LER is obtained by computing the edit distance between result labels from best path decoding and the transcripts, while LAcc is obtained by value 1.0 minus LER. In EESEN, the strategy of learning rate adjustment, called "newbob" (Newbob-Dev), is achieved through LAcc on development set. However, our experiments show that Newbob-Dev strategy might not suit for CTC-trained system. We then adopt a strategy whose adjustment is guided by training set, and call it Newbob-Trn.

Fig. 5 is the basis why we use Newbob-Trn, the legend like DevLAcc means LAcc on development set. As seen in Fig. 5, DevLAcc curves are not stable, which lead to early stopping when using Newbob-Dev. Reasons are that learning rate is halved after each LAcc dropping, it quickly tends to zero in early iterations because of it's small starting value and then the training stops. However, TrnLAcc is stable, so the model can be trained more sufficiently using Newbob-Trn.

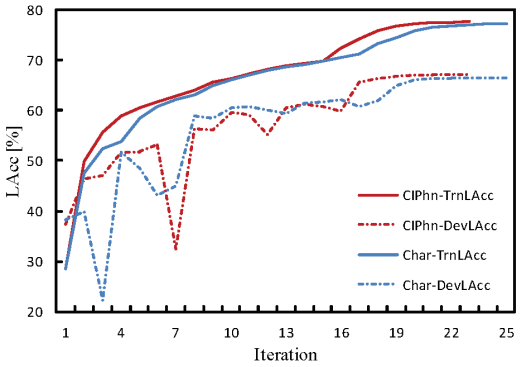

Figure 5: LAcc curves of training set and development set using CTC-trained BiLSTM model whose output units are Chars or CI-Phns.

The reason that Newbob-Dev doesn't take effect might due to the incomplete development set. In a hybrid system, even a small development set include enough frames to cover all types of phoneme. However, in a CTC trained system, one sequence is seen as one sample, so a small development set has little ability to cover most of samples in training set and hence fails to represent training set.

Although Newbob-Trn might lead to over-fitting, in our experiments, LAccs on both training set and development set are convergent, stable and no over-fitting occurs. The following experiments all adopt Newbob-Trn strategy.

4.3. Blank Label Prior Cost

We implement blank label prior costing by scaling it. Table 2 shows our results with different output units when the cost were added. The item Scale means the scaling number, if Scale is 1.0, it is just EESEN's method. The percentages in brackets is

the relative CER reductions compared with EESEN method's results (Scale=1.0).

Table 2: Results when extra cost is added on blank label prior.

Models	Unit	Scale	CER(%)
BiLSTM	Char	1.0	37.16
		0.15	34.22 (7.9%)
	CI-Phn	1.0	36.96
		0.1	33.10 (10.04%)
	CD-Phn	1.0	38.82
		0.03	31.81 (18.1%)

According to Table 2, it's clear that reductions on blank label prior really take good effects on all type of units. Specifically, CD-Phn model has the most obvious improvement, followed by CI-Phn model, finally Char model. We should know that CD-Phns are the most fined-grained units so there are many blank labels in its augmented label sequence. So a strong assumption is that different improvements are associated with different blank label prior values and the prior costing is meaningful and useful.

Through this method, our CD-Phn model outperforms our baseline (BiLSTM-HMM), which means CTC can bring better performance for ASR. Our Char model also performs well. In the published academic papers, the best CER on HKUST from CTC-trained end-to-end system is 39.70% [14] (with chars as output units), while our CER is 34.22%, the improvement is 13.8%.

The following experiments all adopt blank label prior cost.

4.4. UniLSTM with RC

In this section, we show the feasibility of UniLSTM-RC in CTC-trained systems on Mandarin. Table 3 is the experimental results about CTC-trained UniLSTM-RC, where τ is decided by the performance of development set.

Table 3: CTC trained UniLSTM.

Models	Unit	RC	Trn LAcc(%)	Dev LAcc(%)	CER (%)
UniLSTM	Char	No	68.46	61.06	38.39
		τ=15	74.55	64.66	35.34
	CI-Phn	No	68.52	60.22	37.75
		τ=20	74.22	65.03	34.35
	CD-Phn	No	69.76	53.38	32.14
		τ=25	73.94	57.20	30.96

Table 3 shows that RC really profits the pure UniLSTM model on all types of output unit. The relevant improvement on Char units, CI-Phn units and CD-Phn units are 7.9%, 9% and 3.7% respectively. According to the corresponding results in Table 2, UniLSTM with RC is able to match BiLSTM.

[001-085]:SIL	[086-107]:呃	[108-219]:SIL	
[220-243]:我	[244-253]:觉	[254-261]:得	[262-267]:他
[268-284]:挺	[285-301]:好	[302-329]:的	[330-380]:SIL

Figure 6: Force alignment information of a selected utterance.

For further understanding the effect of RC, we randomly select an utterance from development set. The force alignment information obtained by GMM-HMM system between frames and characters is illustrated in Fig. 6. Then, we collect characters' posteriors from trained end-to-end BiLSTM, UniLSTM and UniLSTM-RC models respectively, and figure them. According to the alignment information, the results of Fig. 7 present that all the models have output delays. We see that BiLSTM makes the most accurate position prediction of the six Chinese characters, and prediction similarity between BiLSTM and UniLSTM-RC is higher than the similarity between UniLSTM-RC and UniLSTM. That means RC greatly decreases output delays and UniLSTM-RC is a great substitution of UniLSTM for online CTC-based ASR systems.

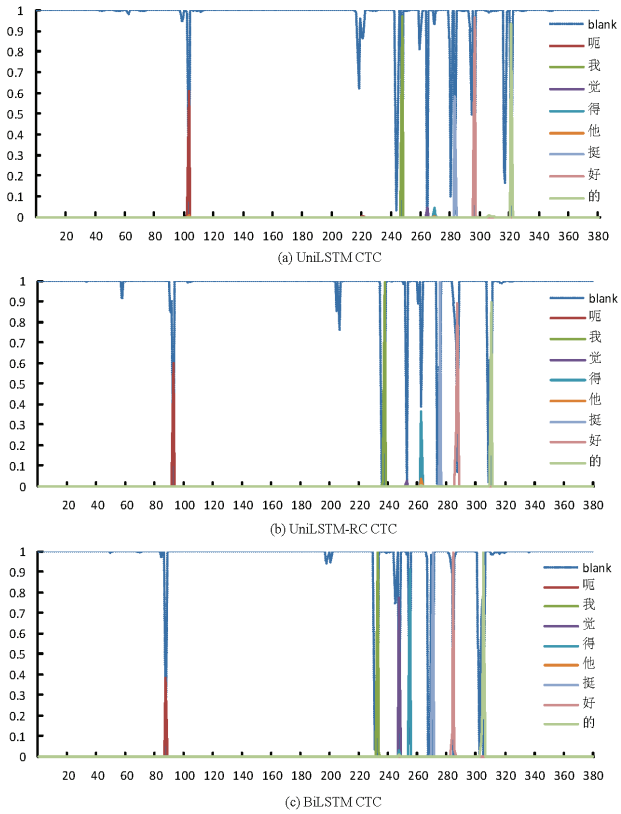

Figure 7: Characters' posterior probability curves from different LSTM models. The horizontal coordinate stands for speech frames, the longitudinal coordinate stands for posterior probability.

5. Conclusions

In this paper, we establish CTC-based systems on Chinese Mandarin ASR task. Three different level output units are explored: Chars, CI-Phns and CD-Phns. We also improve the training strategy and posterior normalization. We propose Newbob-Trn strategy to make training stable and adequate, and add extra cost on blank label prior before normalizeing posterior when decoding. Benefited from CTC and the improvements, our CD-Phn model outperforms the hybrid CD states model. What's more, our end-to-end model (Char model) achieves a rather good performance. Further, we establish the CTC-trained UniLSTM-RC model, which ensures the real-time requirement of a online system, meanwhile, brings performance gain.

In the future, we will focus on pure end-to-end ASR systems, at the same time, accelerate its training and testing processes. Then, we will apply CTC-based end-to-end method to larger vocabulary Chinese Mandarin ASR tasks.

6. References

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[2] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[3] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *INTERSPEECH*, 2014, pp. 338–342.

[4] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "Gmm-free dnn training," in *Proceedings of ICASSP*, 2014, pp. 5639–5643.

[5] M. Bacchiani, A. W. Senior, and G. Heigold, "Asynchronous, online, gmm-free training of a context dependent acoustic model for speech recognition." in *INTERSPEECH*, 2014, pp. 1900–1904.

[6] 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.

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

[8] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

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

[10] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[11] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[12] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4280–4284.

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

[14] Y. Miao, G. Mohammad, X. Na, K. Tom, M. Florian, and W. Alexander, "An empirical exploration of ctc acoustic models," in *Acoustics, Speech and Signal Processing, 2016. ICASSP 2016. IEEE International Conference on*. IEEE, 2016.

[15] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," *arXiv preprint arXiv:1507.08240*, 2015.

[16] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012.

[17] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[18] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[19] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[21] 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.

[22] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," in *Learning Theory*. Springer, 2005, pp. 545–560.

[23] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4520–4524.