INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

On Online Attention-based Speech Recognition and Joint Mandarin Character-Pinyin Training

William Chan[1], *Ian Lane*[1,2]

Carnegie Mellon University
[1]Electrical and Computer Engineering, [2]Language Technologies Institute
williamchan@cmu.edu, lane@cmu.edu

Abstract

In this paper, we explore the use of attention-based models for online speech recognition without the usage of language models or searching. Our model is based on an attention-based neural network which directly emits English/Mandarin characters as outputs. The model jointly learns the pronunciation, acoustic and language model. We evaluate the model for online speech recognition on English and Mandarin. On English, we achieve a 33.0% WER on the WSJ task, or a 5.4% absolute reduction in WER compared to an online CTC based system. We also introduce a new training method and show how we can learn joint Mandarin Character-Pinyin models. Our Mandarin character only model achieves a 72% CER on the GALE Phase 2 evaluation, and with our joint Mandarin Character-Pinyin model, we achieve 59.3% CER or 12.7% absolute improvement over the character only model.

Index Terms: Automatic Speech Recognition, Recurrent Neural Networks, Attention-based Speech Recognition

1. Introduction

Recently, there has been much interest in end-to-end Automatic Speech Recognition (ASR) models [1, 2, 3, 4, 5, 6]. End-to-end ASR models are attractive to many researchers due to its simplicity in training and inference compared to most HMM-based models. However, much of the past work on end-to-end speech models have been focused on offline models (with the exception of [6] for an online CTC model, and [7] for TIMIT). In this paper, we extend on the previous models and explore online end-to-end attention-based ASR models that do not use Language Models (LMs) or searching.

LM-free models are very attractive due to their simplicity during inference. For example, n-gram LMs used in HMM systems for decoding are typically independently trained and large (e.g., $O(1G)$) in size. This limits the deployments of hybrid ASR systems without compacting the LM [8]. The end-to-end model we present is capable of learning the language of the output space directly (however, limited to the number of labelled transcripts).

Search-free models are also attractive due to their simplicity during inference. We do not need to keep track of n-best hypotheses nor its states, which could be advantageous in computing platforms with limited memory, computational capacity and latency requirements.

This paper will focus on end-to-end attention-based models without using LMs or searching during inference. We experiment with both English and Mandarin. A sequence-to-sequence model directly outputing Mandarin characters may be difficult to learn with due to the large vocabulary of the output space. We introduce a joint Mandarin Character-Pinyin model, wherein we can jointly learn the Mandarin character sequence and Pinyin sequence directly. We found by incorporating the additional Pinyin information during training, we can achieve significant reduction in CER. Our method also only uses the Pinyin information during training, and thus during inference, it has the exact same model structure (and runtime cost) as the Mandarin character only model.

2. Related Work

Connectionist Temporal Classification (CTC) [9] speech recognition models were first shown to be capable of directly model acoustics to English characters directly in one neural network model [1]. Subsequent experiments extended on the original work including large scale data [2] and Mandarin characters [6]. However, CTC models have a conditional independence assumption on its output tokens, wherein it will become difficult to model the language of the data (i.e., the interdependencies between words) [1].

Attention based sequence-to-sequence models were recently introduced for speech recognition [3, 4, 5]. Sequence-to-sequence models do not have the Markovian or conditional independence assumption, allowing the model to be very expressive and model long term dependencies between characters. However, most previous work have focused on offline models, wherein the encoder is a bidirectional RNN (with an exception being [7] applied unidirectional attention-based models on TIMIT). It is unclear whether we can use an unidirectional encoder on character based systems, since the attention mechanism can benefit substantially from the lookahead mechanism a bidirectional RNN provides. In this paper, we will attempt to answer this question and show we can use unidirectional RNNs for the encoder (and make the model online), however at some performance loss.

3. Model

We will describe our online attention model in this section. Our model closely relates to the end-to-end neural speech recognizers proposed by [4, 5]. The model consists of two components, an encoder and an attention-based decoder, we will first describe the encoder. Let $\mathbf{x} = (x_0, \dots, x_T)$ be our input audio sequence (e.g., sequence of filter bank spectra). We process the input signal $\mathbf{x}$ into higher level features $\mathbf{h}$ a with an unidirectional encoder RNN:

$$\mathbf{h} = \mathrm{RNNEncoder}(\mathbf{x}) \tag{1}$$

where RNNEncoder is an unidirectional RNN network. We use a deep GRU network with hierarchical subsampling [10, 4,

http://dx.doi.org/10.21437/Interspeech.2016-334

5] for the $\mathrm{RNNEncoder}$ function. The hierarchical subsampling allows the attention model (described below) to attend to fewer timesteps and help prevent the attention alignment from being diluted. It also helps speeds up the training and inference computation since a single frame of h can span over many frames of x. See Figure 1 for visualization.

The decoder network generates a character sequence $\mathbf{y} = (y_0, \ldots, y_U)$ with an attention-based RNN:

$$s_j = \mathrm{RNNDecoder}(y_{j-1}, s_{j-1}, c_{j-1}) \tag{2}$$

where s_j is the state of the decoder RNN and $\mathrm{RNNDecoder}$ is a GRU network in our experiments. The context c_j is generated by an $\mathrm{AttentionContext}$ mechanism from the RNN decoder state s_j and some encoder features $\mathbf{w}_j$:

$$c_j = \mathrm{AttentionContext}(s_j, \mathbf{w}_j) \tag{3}$$

For the AttentionContext function, we use a MLP attention mechanism [11]. First, we compute the energies $e_{i,j}$, then the normalized alignments $\alpha_{i,j}$ between the decoder state s_j and encoder features $\mathbf{w}_j$. The context c_j is the weighted bag of features over $\mathbf{w}_j$ using the alignments α as the weights:

$$e_{j,i} = \langle v, \tanh(\phi(w_i) + \psi(s_j) + b)\rangle \tag{4}$$

$$\alpha_{j,i} = \frac{\exp(e_{j,i})}{\sum_i \exp(e_{j,i})} \tag{5}$$

$$c_j = \sum_i \alpha_{j,i} w_i \tag{6}$$

where ϕ and ψ are MLP networks and v, b are weight vectors. If $\mathbf{w}_j = \mathbf{h}$ [11, 4], then we have our standard attention-based sequence-to-sequence model, and the model is not decodable online as we need to wait for the entire input acoustic signal to be seen before we can begin decoding. We follow an approach similar to [5] and use a sliding window, we use the median of the previous alignment α_{j-1} to create a sliding window $\mathbf{w}_j$:

$$m_j = \mathrm{median}(\alpha_{j-1}) \tag{7}$$

$$\mathbf{w}_j = \{h_{m_j - p}, \ldots, h_{m_j + q}\} \tag{8}$$

where m_j is the median of the previous alignment, and p, q are the hyperparameters to our window size. In our experiments we set $p = 100$ and $q = 10$. This means, we can start decoding (and continue decoding) as long as the we have up till $m_j + q$ frames of h signal available (since the $\mathrm{RNNEncoder}$ function is unidirectional).

The model produces a conditional distribution as a function over all previously emitted characters and the sliding window of acoustic features $\mathbf{w}_j$:

$$p(y_j | \mathbf{w}_j, y_{<j}) = \mathrm{softmax}(\varphi(s_j, c_j)) \tag{9}$$

where φ is a MLP. We note that our model will likely have difficulties predicting the next token if there is a large gap or silence between characters, for example if there was silence for more than q frames. We however found this to not be a problem with the datasets we experiment with. We also note the model in Equation 9 is non-Markovian, meaning we can learn the language directly (which is much harder for CTC systems due to its conditional independence assumption). However (as noted by [4]), the implicit language model learnt is limited by the number of transcribed transcripts unlike n-gram or RNN LMs which can leverage on any text data. See Figure 2 for visualization of our sliding window model.

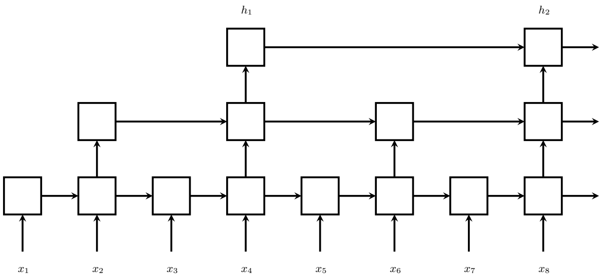

Figure 1: Hierarchical recurrent neural network: we subsample the inputs $\mathbf{x}$ to reduce the time dimension into higher level features $\mathbf{h}$.

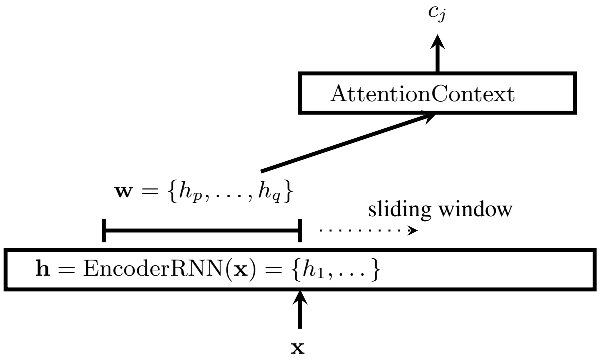
c_j
AttentionContext
$\mathbf{w} = \{h_p, \ldots, h_q\}$
sliding window
$\mathbf{h} = \mathrm{EncoderRNN}(\mathbf{x}) = \{h_1, \ldots\}$
$\mathbf{x}$

Figure 2: Online Attention: The acoustic signal $\mathbf{x}$ are processed online by an unidirectional RNN into features $\mathbf{h}$. A sliding window $\mathbf{w}_j$ as a function of the previous alignment moves across $\mathbf{h}$ to be processed by the attention $\mathrm{AttentionContext}$ mechanism for the next alignment.

4. Optimization

We optimize the parameters of our model to maximize the log probability of the correct sequences:

$$\max_\theta \sum_j \log P(y_j | \mathbf{w}_j, \tilde{y}_{<j}; \theta) \tag{10}$$

where $\tilde{y}_{i-1}$ is the ground truth previous character or a character randomly sampled (with 10% probability) from the model using the sampling procedure from [12, 4]. We found the sampling technique to help reduce overfitting.

5. Decoding

Unlike CTC systems, neural attention models can learn the language of our output tokens due to the conditional dependency of previously emitted symbols $y_{<j}$ in Equation 9. As also observed in [4, 5], while LMs and beam search can improve results, the improvement is small relative to CTC and HMM based systems. Thus, in this paper we focus on experiments that do not use any LMs or searching. We simply take the greedy path of our conditional model in Equation 9:

$$\hat{y}_j = \mathrm{argmax}_{y_j} P(y_j | \mathbf{w}_j, y_{<j}) \tag{11}$$

We believe many applications will benefit from models that do not use searching, for example on embedded platforms without GPUs and limited computation and memory capacity.

6. Joint Mandarin Character-Pinyin Model

We found the attention model difficult to converge with Mandarin data. The attention mechanism has a difficult time learning the alignment and/or the decoder overfits to the transcripts before the acoustic model encoder converges. We suspect this is due to Mandarin using a logographic orthography and the Chinese characters give limited information on the sounds of the spoken language. Additionally, the large vocabulary and the conditional dependency of our model (unlike CTC which is conditionally independent [13, 14]) makes learning a generalized model more difficult.

Mandarin characters can be readily transcribed to Pinyin, which is its the romanization and a (rough) phonetic representation with English characters using a Mandarin Pinyin dictionary. The cost to transform the Mandarin characters to Pinyin is minimal and only consists of a lookup. While we could build our ASR system to directly output Pinyin, such a system is nonideal as we would require another system to convert the Pinyin (with possible mis-spellings) back into the Mandarin characters. We want our model to leverage the phonetic Pinyin information readily available while still emitting Mandarin characters as our output.

We adapt our network architecture to jointly learn the Character and Pinyin transcriptions, and discard the Pinyin paths during inference (to be explicit, our model still only conditions on the Mandarin characters and never conditions on the Pinyin). We extract the Pinyin transcriptions using a dictionary [15] (without the tones). Each Mandarin character has a phonetic Pinyin with a maximum Pinyin character length of 7. We pad each Pinyin representation to be exactly length 7. Thus, for each Mandarin character y_j, we can lookup its Pinyin transcription $\mathbf{z}_j = \mathrm{PinyinDictionary}(y_j)$ with each vector $\mathbf{z}_j = (z_1, \ldots, z_7)$. For Out-of-Vocabulary (OOV) words in the Pinyin dictionary, we use the original character $z_{j,1} = y_j$. The Pinyin dictionary [15] models bigram of Mandarin characters, we thus greedily use these transcriptions if available. Additionally, the dictionary may give multiple Pinyin transcriptions for a Mandarin character, in this case we simply use the first transcription in the dictionary.

We model the Pinyin characters with additional MLP networks as a function of the $\mathrm{RNNDecoder}$ and attention states:

$$\log p(\mathbf{z}_j | \mathbf{w}_j, y_{<j}) = \sum_k \log \mathrm{softmax}(\varphi_k(s_j, c_j)) \quad (12)$$

where y_j are the Mandarin characters and φ_k are MLPs modelling the Pinyin characters.

We jointly optimized the model to learn the Mandarin character and the Pinyin representation. We do not condition on the Pinyin characters, thus we only use the Pinyin information during training. During inference we discard the Pinyin outputs and only read off the Mandarin softmax outputs. The joint Mandarin Character-Pinyin model optimization objective is the joint probability of the Mandarin characters y_j and Pinyin $\mathbf{z}_j$:

$$\max_\theta \sum_i \log P(y_j | \mathbf{w}_j, \tilde{y}_{<i}; \theta) + \log P(\mathbf{z}_j | \mathbf{w}, \tilde{y}_{<i}; \theta) \quad (13)$$

The motivation of providing the Pinyin information is not to give the model perfect pronunciation information, but rather to give it some (possibly noisy) information on how each Mandarin character is pronounced. We hypothesize that this additional pronunciation information can be backpropagated to the encoder and learn a more generalized model. Without the additional Pinyin information, we hypothesize the decoder would very easily overfit to the training transcripts without using the acoustics. We leverage on the explicit alignment (as opposed to implicit alignment of CTC) to inject this additional pronunciation information into the model. We also hypothesize that if we had sufficient data, we would not need this technique [2, 4].

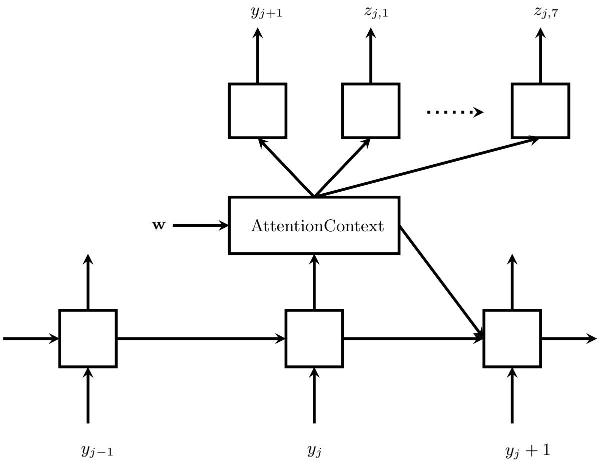

Figure 3: Joint Mandarin Character-Pinyin decoder: we model the Mandarin character y_j and Pinyin $\mathbf{z}_j = (z_{j,1}, \ldots, z_{j,7})$ together in the decoder RNN. The sliding window $\mathbf{w}$ of encoder features is from Figure 2. The $\mathrm{AttentionContext}$ function consumes the window $\mathbf{w}$ to produce a context to emit the character or Pinyin outputs and update the RNN state.

7. Experiments

We experimented with English Wall Street Journal (WSJ) and GALE Mandarin datasets. We tried to use similar hyperpameters and models between the two datasets, however there are some minor differences as described below.

First, we used AdaDelta [16], we found it converges much more quickly than Stochastic Gradient Descent (SGD) even with acceleration [17]. We used AdaDelta with $\rho = 0.95$ and $\epsilon = 1e - 8$ to optimize our model, however we may lower the value of ϵ as training progresses and described in the later subsections below. Despite AdaDelta's decay, we found resetting (i.e., zero out) the AdaDelta accumulators $\mathbb{E}[g^2]$ and $\mathbb{E}[\Delta x^t]$ after each epoch to help with the optimization. We clipped the gradients to have a maximum norm of 1.

We initialized all the weighted connection matrices of our model with uniform distribution $\mathcal{U}(-0.1, 0.1)$, and for our embedding matrix we used a uniform distribution $\mathcal{U}(-\sqrt{3}, \sqrt{3})$. We initialized the GRU update and reset gate bias to one and all other biases to zero. We imposed a max column norm of 1 for all our weight matrices after each update [18].

If our $\mathrm{EncoderRNN}$ is a hierarchical RNN, then our model may overfit towards the subsampling permutations of our architecture. We follow an approach typically found in computer vision [19], we randomly add up to f frames of input delay to our acoustic signal, where f is the subsampling factor of the $\mathrm{EncoderRNN}$ (e.g., $f = 4$ for English WSJ).

7.1. Wall Street Journal

We first ran experiments on Wall Street Journal (WSJ) (available as LDC93S6A and LDC94S13A). We used the si284 as

Table 1: Wall Street Journal WERs: We train end-to-end WSJ models without LMs or searching. Compared to online CTC, our attention model has a 17% relative reduction in WER.

Model	eval92 WER
Hybrid-HMM Models	
DNN-HMM [20]	3.8
RNN-HMM [21]	3.5
Offline Character Models	
CTC (Graves et al., 2014) [1]	30.1
CTC (Hannun et al., 2014) [22]	35.8
Attention + Conv (Bahdanau et al., 2016) [5, 23]	21.3
Attention + TLE (Bahdanau et al., 2015) [23]	18.8
Online Character Models	
CTC (Hwang et al., 2016) [6]	38.4
Online Attention (this work)	33.0

the training set, dev93 as the validation set and eval92 as the test set. We observe the WER of the validation set after epoch and stop training when the validation WER no longer improves. We used 40 dimensional filterbanks with energies, delta and delta-delta coefficients (total 123 dimensional features). We follow [1] in text normalization, our token output space consists of the English alphabets, space and punctuations.

We used 384 GRU cells for the encoder, with 3 layers, and the last two layers being hierarchical subsampling layers with a factor of two (i.e., in total the encoder reduces the frame rate by $4 = 2^2 2$). The decoder is a 2 layer attention-based RNN with 256 GRU cells. We trained the model for around 30 epochs, after 10 epochs we manually lowered the AdaDelta ϵ several times manually to a final value of $\epsilon = 1e - 15$.

Table 1 gives the WER of our online attention model as well as a comparison to several other models. First, our online attention model achieves a 33.0% WER without LMs or searching. This compares to 21.3% WER without LMs or searching trained on cross entropy or 18.8% WER trained with Task Loss Estimation (TLE) [23]. It is clear the offline networks performs much better for this task, we also note that [5, 23] has a convolutional location-based attention mechanism which our model does not have. Our online attention model however does outperform online CTC models [6] which achieved 38.4% WER or a 5.4% absolute improvement.

However, compared to the state-of-the-art HMM models [21] (with trigram LMs), the online end-to-end ASR models are still significantly behind. We believe more data would ameliorate this issue, as the attention-based end-to-end models are much more powerful and overfit much more easily [4].

7.2. GALE Mandarin

We also experimented with the GALE Phase 2 Chinese Broadcast News Speech (LDC2013S08 and LDC2013T20). We used the exact same train (about 104 hrs) and test (about 6 hrs) split as in the Kaldi s5 recipe [20]. We used 40 dimensional filterbanks with energies as our input features. For the encoder, we used 384 GRU cells with 2 layers. We used hierarchical subsampling with a factor of 2 for the second layer only. We also used "super-frames" [24] stacking 16 frames with a stride of 4. The combination of the subsampling and superframes greatly decreased our encoder feature dimensions and consequently our training time as well. The decoder is a 1 layer attention-based RNN with 256 GRU cells. We trained the model for approximately 20 epochs until convergence.

Table 2: GALE Mandarin CERs: We train end-to-end Mandarin models without LMs or searching. The Pinyin model uses the Pinyin phonetic information during training, but discards it during inference.

Model	CER
DNN-HMM [25]	18.5
Online Attention Character	72.0
Online Attention Character + Pinyin	59.3

Table 2 gives an overview of our experimental results in CER. We achieved a 72% CER in our online end-to-end mandarin model, this compares to 18.5% CER for a Kaldi DNN-HMM system. We spent considerable effort in hyperparameter tuning including optimization and regularization hyperparmeters, however we were unable to improve the results of the Mandarin only model. We suspect the large vocabulary (around 4000 Chinese characters) with relatively limited training data [4] made the mode difficult to learn and generalize well.

We trained a joint Character-Pinyin model and the benefits of adding the Pinyin information into the training process was obvious. We achieved a CER of 59.3% for our joint Mandarin Character-Pinyin or 12.7% absolute improvement over the character only model. Mandarin characters generally have limited phonetic information (i.e., the strokes), especially since our input is Chinese characters (as opposed to the strokes or pixels of the characters) whereas the Pinyin transcription more closely matches the phonetic representation of Mandarin speech. The neural network model is able to leverage this additional phonetic information. While we do use the additional Pinyin information during training, our model does not condition on the Pinyin and thus has the exact same model architecture (and inference cost) as the Mandarin Character only model.

Finally, without any attempt on minimizing the model size (e.g., SVD decomposition, projection layers or quantization [26, 27]) we note that both are English and Mandarin models are less than 64 MiB in size. We believe future research in this area can lead to extremely compact ASR models applicable for embedded devices.

8. Conclusions

In this paper, we presented an online end-to-end speech recognition model based on attention-based neural networks. Our models do not rely on any pronunciation dictionaries, language models or searching during inference. On English, our model achieves a WER of 33.0% on WSJ, or an 5.4% absolute improvement over an online end-to-end CTC system [6]. On Chinese Mandarin models, we show how to build a joint Character-Pinyin model, combining the available Pinyin transcriptions to our end-to-end character model. The joint model does not condition on the extra Pinyin information and is only used during training. For the GALE Phase 2 Chinese Broadcast News Speech evaluation, we achieved a CER of 72.0% for our character only model. With our joint Character-Pinyin model, we achieve a CER of 59.3% or a 12.7% CER improvement over the character model only.

9. References

[1] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *International Conference on Machine Learning*, 2014.

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng, "Deep Speech: Scaling up end-to-end speech recognition," in *arXiv*, 2014.

[3] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent nerual network encoder-decoder for large vocabulary speech recognition," in *INTERSPEECH*, 2015.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[5] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[6] H. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[7] N. Jaitly, Q. Le, O. Vinyals, I. Sutskever, and S. Bengio, "An Online Sequence-to-Sequence Model Using Partial Conditioning," in *arXiv*, 2015.

[8] I. McGraw, R. Prabhavalka, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, and C. Parada, "Personalized Speech Recognition on Mobile Devices," in *arXiv*, 2016.

[9] A. Graves, S. Fernandez, F. Gomez, and J. Schmiduber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *International Conference on Machine Learning*, 2006.

[10] S. Hihi and Y. Bengio, "Hierarchical Recurrent Neural Networks for Long-Term Dependencies," in *Neural Information Processing Systems*, 1996.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *International Conference of Learning Representations*, 2015.

[12] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks," in *Neural Information Processing Systems*, 2015.

[13] J. Li, H. Zhang, X. Cai, and B. Xu, "Towards end-to-end speech recognition for chinese mandarin using long short-term memory recurrent neural networks," in *INTERSPEECH*, 2015.

[14] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, "An emprical exploration of CTC acoustic models," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[15] "CC-CEDICT," cc-cedict.org, 2016.

[16] M. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," in *arXiv*, 2012.

[17] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," in *International Conference on Machine Learning*, 2013.

[18] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing coadaptation of feature detectors," in *arXiv*, 2012.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Neural Information Processing Systems*, 2012.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannenmann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Automatic Speech Recognition and Understanding Workshop*, 2011.

[21] W. Chan and I. Lane, "Deep Recurrent Neural Networks for Acoustic Modelling," in *arXiv*, 2015.

[22] A. Hannun, A. Maas, D. Jurafsky, and A. Ng, "First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs," in *arXiv*, 2014.

[23] D. Bahdanau, D. Serdyuk, P. Brakel, N. R. Ke, J. Chorowski, A. Courville, and Y. Bengio, "Task Loss Estimation for Sequence Prediction," in *arXiv*, 2015.

[24] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition," in *INTERSPEECH*, 2015.

[25] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for Speech Recognition," in *INTERSPEECH*, 2015.

[26] V. Vanhoucke and A. Senior, "Improving the speed of neural networks on CPUs," in *Neural Information Processing Systems: Deep Learning and Unsupervised Feature Learning Workshop*, 2011.

[27] Y. Wang, J. Li, and Y. Gong, "Small-footprint high-performance deep neural network-based speech recognition using split-vq," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.