Boosting Character-Based Chinese Speech Synthesis via Multi-Task Learning and Dictionary Tutoring

Yuxiang Zou[1,2], *Linhao Dong*[1,2], *Bo Xu*[1]

[1]Institute of Automation, Chinese Academy of Sciences, China
[2]University of Chinese Academy of Sciences, China
{zouyuxiang2017, donglinhao2015, xubo}@ia.ac.cn

Abstract

Recent character-based end-to-end text-to-speech (TTS) systems have shown promising performance in natural speech generation, especially for English. However, for Chinese TTS, the character-based model is easy to generate speech with wrong pronunciation due to the label sparsity issue. To address this issue, we introduce an additional learning task of character-to-pinyin mapping to boost the pronunciation learning of characters, and leverage a pre-trained dictionary network to correct the pronunciation mistake through joint training. Specifically, our model predicts pinyin labels as an auxiliary task to assist learning better hidden representations of Chinese characters, where pinyin is a standard phonetic representation for Chinese characters. The dictionary network plays a role as a tutor to further help hidden representation learning. Experiments demonstrate that employing the pinyin auxiliary task and an external dictionary network clearly enhances the naturalness and intelligibility of the synthetic speech directly from the Chinese character sequences.

Index Terms: Chinese speech synthesis, multi-task learning, dictionary tutoring

1. Introduction

Text-to-speech (TTS) [1] systems convert normal language text into human speech, aiming to synthesize speech with high intelligibility and naturalness. Compared with statistical parametric speech synthesis (SPSS) [2] [3] [4] which has a text frontend extracting various linguistic features, sequence-to-sequence (seq2seq) [5] neural TTS [6] [7] [8] [9] has become a new trend due to its simpler module and procedure and less need for extensive domain expertise. Besides, end-to-end neural TTS can generate more natural and human-like speech than traditional TTS [10].

Seq2seq attention-based models have achieved promising results on English text-to-speech tasks and the use of characters brings model simplicity and enables end-to-end optimization. Char2Wav [11] first explores end-to-end attention-based model trained on characters for TTS tasks. Later, Wang et al. propose Tacotron [7], an end-to-end generative text-to-speech model that synthesizes speech directly from characters. On the basis of Tacotron, Tacotron2 [10] combines a Tacotron-style model and a modified WaveNet [12], thus it is able to generate sound much closer to natural human speech from character sequences. Since character-based TTS models do not require a grapheme-to-phoneme conversion model [13], modeling with character can alleviate the need of manual labeling cost and is becoming a trend in end-to-end TTS model.

Although character-based models perform well on English TTS, it is still a big challenge for Chinese TTS due to the severe label sparsity issue [14]. Unlike English, Mandarin Chinese has tens of thousands of characters. Limited to the amount of training data, words with few occurrences in training corpus cannot be fully trained and easily have a strange prosody and even wrong pronunciation.

In this paper, we propose a multi-task learning [15] and dictionary tutoring method to address the issue of label sparsity. Specifically, We introduce an additional task of character-to-pinyin mapping to assist TTS learning. Pinyin is a standard phonetic representation for each character, which is a smaller modeling unit compared with character, therefore, learning speech features jointly with character-to-pinyin mapping can integrate more delicate information. Since the pronunciation problem cannot be completely solved by the multi-task method, we additionally introduce a dictionary network. The dictionary network also learns a character-to-pinyin mapping, but pre-trained on much larger dataset. When the predicted pinyin confidence of multi-task learning is not high enough, the pinyin embedding is substituted by the dictionary embedding. The whole process is like a language beginner looking up the dictionary when encountering a word that he does not know how to pronounce during reading. Our main contributions can be summarized as follows:

- We first use three different modeling units, including character, pinyin and phoneme, to synthesize Chinese speech based on Tacotron2 in order to analyze effects of modeling units on Chinese TTS task, as well as to create baselines.
- We propose a multi-task learning method to address the issue of label sparsity, which enables generating more natural and intelligible speech.
- We demonstrate that dictionary tutoring mechanism has the ability to correct pronunciation mistakes of uncommon and polyphonic characters to some degree.
- Our model can synthesize speech directly from Chinese characters.

2. Related work

To our knowledge, sequence-to-sequence attention-based models perform very well on English TTS tasks, nevertheless, related works are quite few on Mandarin Chinese TTS tasks. Wang et al.[16] is the earliest work touching end-to-end TTS using attention mechanism in Chinese TTS. It is trained on untoned phoneme input and the experimental results seem to be somewhat limited. In [17], a forward attention method based on Tacotron is proposed for Chinese TTS, which learns the mono-

The research work is supported by the Key Research and Development Program of Beijing (No.Z181100008918017).

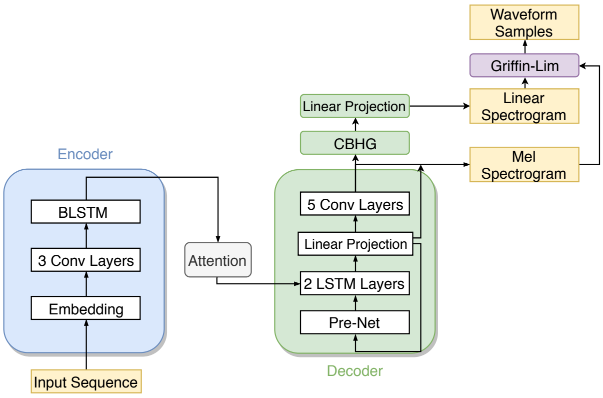


Figure 1: *The architecture of baseline TTS system.*

tonic alignment from phone sequences to acoustic sequences and improves naturalness of synthetic speech.

Those training data both require phoneme labels which are difficult to get and laborious to mark. Additionally, those methods need an additional front-end grapheme-to-phoneme conversion module which increases speech synthesis pipeline complexity. Different from those, our proposed model directly uses Chinese characters as input and thus simplifies the pipeline. Different from [16], our model is an end-to-end optimization model and does not need to predict vocoder parameters.

3. Proposed approach

3.1. Baseline neural TTS system

In this section, we describe our baseline neural text-to-speech system. Figure 1 illustrates the network structure of our slightly modified baseline based on Tacotron2 [10], which is a sequence-to-sequence architecture that consists of encoder and decoder with attention mechanism. The encoder maps an input text sequence $\boldsymbol{x} = (x_1, ..., x_L)$ to a series of hidden representations which are consumed by the decoder to predict a spectrogram sequence $\boldsymbol{y} = (y_1, ..., y_T)$. Finally, the Griffin-Lim algorithm [18] is employed to synthesize the waveform from the predicted spectrogram.

On the encoder side, the encoder takes a token sequence as input and passes it through a block of 3 convolutional layers followed by a bi-directional LSTM layer. The hidden states of the recurrent network are used as the encoder representations. On the decoder side, the encoder output representations are consumed by LSTM decoder with an attention network. After convolutional post-net predicts log mel spectrograms, we add another post processing network which is used to predict linear scale spectrograms. This post processing network is simply a CBHG block in Tacotron [7] followed by a linear projection layer.

The loss function in baseline TTS system is the sum of folowing two parts. One is the mel loss and the other is the linear loss. The mel loss L_{mel} is the summed mean squared error (MSE) between the ground truth and the prediction of log mel spectrogram. The linear loss L_{linear} is the first norm between the ground truth and the prediction of linear spectrogram. Specially, we increase the proportion of low frequency components of the linear spectrogram to learn more speech information. L2 regularization is also applied to the loss function.

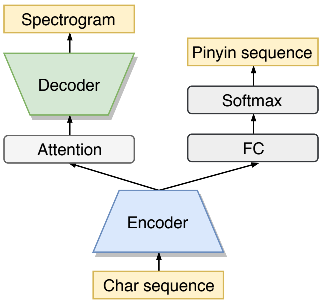


Figure 2: *The architecture of multi-task learning in TTS system. The auxiliary task of character-to-pinyin mapping (on the right) is introduced to baseline TTS.*

$$L_{mel} = \sum_{n=1}^{N} \left(p\left(\boldsymbol{y}_{mel,n} | \boldsymbol{x}_n; \theta\right) - \boldsymbol{y}_{mel,n}^{'} \right)^2 \tag{1}$$

$$L_{linear} = \sum_{n=1}^{N} \left| p\left(\boldsymbol{y}_{linear,n} | \boldsymbol{x}_n; \theta\right) - \boldsymbol{y}_{linear,n}^{'} \right| \tag{2}$$

where N is the number of $\langle text, audio \rangle$ pairs, $\boldsymbol{x}$ is the input text sequence, $\boldsymbol{y}$ is the predicted speech features and $\boldsymbol{y}'$ is the ground truth speech features. The total loss L_{base} of the baseline system is calculated as:

$$\min_{\theta} L_{base} = L_{mel} + L_{linear} \tag{3}$$

where θ represents model parameters.

3.2. Multi-task learning in TTS system

Multi-task learning (MTL) has been used successfully across all applications of machine learning, from computer vision [19] to natural language processing [20] to automatic speech recognition [21]. MTL improves generalization by leveraging the domain-specific information contained in the training signals of related tasks [15]. For our character-based TTS task which suffers from label sparsity issue, we introduce an auxiliary task to learn a character-to-pinyin mapping. We argue that pinyin is a smaller modeling unit compared with character and the auxiliary task is quite relative to our main task, thus it can help our model focus more differentiable features.

The proposed multi-task learning system shown in Figure 2 consists of a main task and an auxiliary task. The main task is to predict spectrogram features from sequences of input character using an acoustic feature prediction network, as described in section3.1. The auxiliary task is to learn pronunciation embedding from character-to-pinyin mapping. In the auxiliary branch, the output of the encoder is also delivered to a single fully connected layer which projects the hidden representations into the pinyin sequence. After a softmax layer, each unit outputs a likelihood probability which represents the confidence level of pinyin prediction. The loss of the auxiliary task is cross entropy (CE) between the pinyin label and the outputs of softmax layer. The total loss for the multi-task TTS model is defined as a weighted sum of the losses propagated from both branches:

$$\min_{\theta} L_{multi-task} = \lambda L_{base} + (1 - \lambda) L_{CE} \tag{4}$$

where L_{base} is defined as in Equation (3) and λ is the weight hyperparameter.

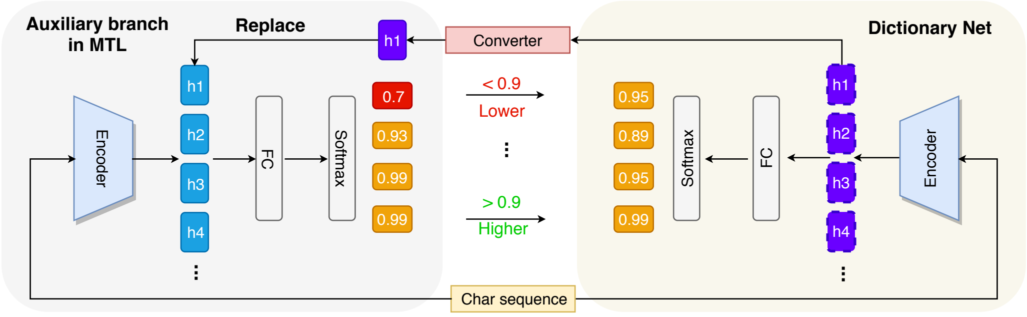


Figure 3: *Illustration of dictionary tutoring. The left side shows the auxiliary branch in multi-task learning. The right side illustrates the pre-trained dictionary network which maps character sequence to pinyin sequence. Since the confidence level 0.7 is lower than the threshold, the "blue" hidden presentation is replaced by the "purple" hidden presentation.*

3.3. Dictionary tutoring in TTS system

With limited amounts of training data, characters with few occurrences in training corpus may have poor coverage. Although we have added an auxiliary task, the encoder may still have the chance of learning bad hidden representations for uncommon character or polyphonic character in Chinese. Meanwhile, the predictions of auxiliary task may be inaccurate, reflected on low confidence level of corresponding softmax outputs. This will lead to wrong pronunciation of synthesized speech.

In order to correct wrong pronunciation, we propose to exploit rich textual knowledge contained in a pronouncing dictionary which typically contains large word and phrase data in the <character, pinyin> pairs. The dictionary network is pre-trained on a large Chinese dictionary, whose structure is the same as the auxiliary branch in the multi-task learning as described in section3.2. After the dictionary network is pre-trained and the multi-task model is trained, we use joint training method to fuse the dictionary information. We call this process dictionary tutoring: as shown in Figure 3, when the predicted pinyin confidence level in the auxiliary branch is lower than threshold we set, the encoder output embedding is replaced by dictionary encoder output embedding. It is worth noting that during dictionary network pre-training, the encoder representation is only supervised by the pinyin label; while during multi-task TTS learning, it is additionally supervised by the speech acoustic feature label, therefore, there exists significant vector space mismatch between the TTS encoder output and the dictionary encoder output. In order to reduce this mismatch, we add a conversion module which is a single fully connected layer. In practice, we fine-tune the added conversion layer during joint training while keeping other parameters frozen.

4. Experiments

4.1. Experimental setup

An open high-quality Mandarin Chinese dataset [22] recorded by a female professional speaker is used in our experiments. This dataset consists of about 12 hours of speech data. The text scripts are in the general domain, covering all kinds of news, novels, science and technology, entertainment, dialogue and other fields. The speech waveform is downsampled to 24kHz from 16-bit mono-channel PCM audio at 48kHz. The database contains 10000 utterances, divided into a training set and a development set, which has 9500 and 500 utterances respectively.

For all experiments, the target acoustic features are log magnitude spectrogram and linear-scale spectrogram extracted with Hann windowing, 50 ms frame length, 12.5ms frame shift and then Griffin-Lim algorithm [18] is used to synthesize waveform from the predicted spectrogram. We train our model with a batch size of 32 by the Adam optimizer [23] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-6}$. The learning rate exponentially decays from 10^{-3} starting after 50,000 steps. Reduction factor is set to 3 for all experiments, i.e., the decoder predicts 3 spectrogram frames at each decoding step.

We first build a baseline neural TTS system using character as input named character-based model. We also experiment with pinyin and phonemes as input to demonstrate how challenging it is to synthesize speech directly from character. The basic architecture and parameters of the three systems are the same while the only difference is that we take different input tokens.

For multi-task learning, the auxiliary network is added to the character-based TTS baseline. The hidden size in the last fully connected layer of the auxiliary task is 1540, the same as the pinyin vocabulary size. The network structure and parameters of the main task are consistent with the baseline system. The weight hyperparameter λ is 0.5.

For dictionary tutoring, the dictionary network is pre-trained on the CMU pronouncing dictionary [24] containing about 110,000 words or phrases with the text form of <character, pinyin> pairs. On the basis of the multi-task model, we only fine-tune the conversion layer while keeping other parameters frozen. The threshold of confidence level is set to 0.9 and a fixed learning rate of 10^{-4} is used during fine-tuning.

4.2. Naturalness test

We perform AB preference tests in terms of naturalness to assess the performances of different systems [1]. 10 native listeners with no hearing difficulties participated in the evaluation using headphones. Each listener evaluated 20 pairs of utterances synthesized from the two comparative systems. After listening to each pair of synthesized utterances, the listeners were asked to choose their preferred one; they could choose "neutral" if they had no preference.

4.2.1. Results on different modeling units

We first compare the performance of character, pinyin and phone based TTS systems. The results are shown in Figure 4. As we can see from the first two lines, both pinyin and phoneme

[1] Audio samples available on https://sysuzyx.github.io/ChineseTTS/

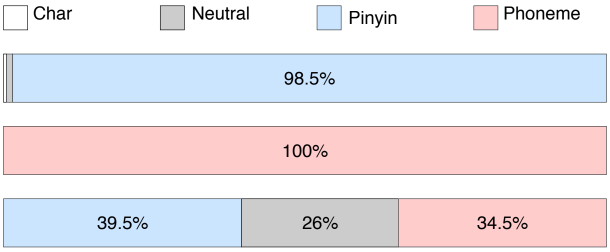


Figure 4: *AB preference results among the three systems modeled with different modeling units, including characters, pinyin and phonemes.*

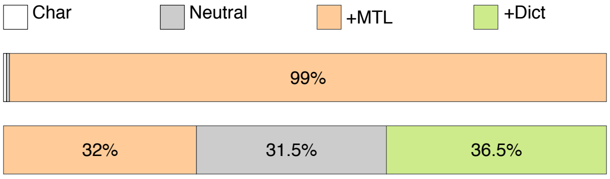


Figure 5: *AB preference results among character-based TTS system, multi-task learning TTS system and dictionary tutoring system.*

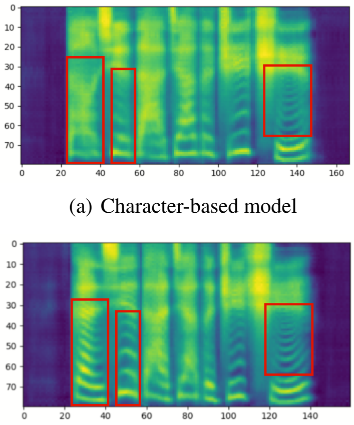
based model significantly outperform character-based model: in both tests, raters strongly preferred them over the characters based baseline by more than 98%. Feedback from raters indicates that speech synthesized by character-based system has a strange tone and prosody and is difficult to understand. This is mainly because the size of characters vocabulary is far larger than the size of pinyin and phoneme vocabulary, and thus the tail characters have a very poor coverage. Besides, the raters considered pinyin-based system against phoneme-based system similarly preferable. Since pinyin is a more intuitive and effective representation for Chinese character, we introduce pinyin instead of phoneme as auxiliary information to the multi-task system as mentioned in section3.2. Overall, The results of AB preference tests demonstrate that it is challenging for building a character-based Chinese text-to-speech model to generate natural speech like human.

4.2.2. Results on multi-task learning and dictionary tutoring

We first compare the performance of our multi-task learning model with character-based TTS system. The result is shown in the first line of Figure 5. It is clear to see that the multi-task learning model we proposed significantly exceeds the baseline character-based system. The results of AB preference tests demonstrate that introducing pinyin labels for multi-task learning can guide the model learning precise pronunciation and significantly improve the naturalness of synthesized speech.

Then, we conduct a similar experiment to compare the performance of dictionary tutoring system with multi-task learning system. The result shown in the last line of Figure 5 demonstrates that the additional dictionary tutoring method has no loss to the naturalness of the synthesized speech. Meanwhile, it can correct wrong pronunciation which will be demonstrated in section4.4.

We also select mel spectrograms generated by the character-based TTS model and the multi-task TTS model respectively with the same text. As shown in Figure 6, the red rectangles in Figure 6(b) contain more delicate spectrogram information compared with that in Figure 6(a). This also confirms that our multi-task system can synthesize clearer speech than character-based model.



(a) Character-based model

(b) Multi-task model

Figure 6: *Mel spectrogram comparison between Character-based model and Multi-task model. Red rectangles are used to mark the differences between two mel spectrograms. Our multi-task model does better in reconstructing details.*

4.3. Case study

In order to verify whether dictionary tutoring has the ability to correct pronunciation mistakes, we conduct case studies by comparing dictionary tutoring system with multi-task learning system. We find dictionary tutoring can correct the tone of polyphonic characters and the pronunciation ambiguity of uncommon characters. For example, ”插曲” is rightly pronounced as *chā qǔ* (in pinyin form) in dictionary tutoring while is wrongly pronounced as *chā qū* in MTL synstem. ”耄耋” is rightly pronounced as *mào dié* in dictionary tutoring while intelligibly pronounced in MTL system (comparison examples can be found on demo page). We can get the conclusion that the added dictionary tutoring method gives the system a certain ability to correct pronouncing mistakes and improves the system robustness.

5. Conclusions

In this paper, we propose a novel and simple method to boost character-based end-to-end Chinese TTS system. Multi-task learning method assists the model learning better by supplementing pinyin domain information. Dictionary tutoring method leverages external rich dictionary information to correct the pronunciation of polyphonic characters and uncommon characters in Chinese. Experimental results show that introducing the pinyin auxiliary task and an external dictionary network clearly enhances the naturalness and intelligibility of the synthetic speech directly from the Chinese character sequences.

6. Acknowledgements

We would like to thank Yiming Wan for his generous help on this work.

7. References

[1] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[2] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *speech communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7962–7966.

[4] S. King, "An introduction to statistical parametric speech synthesis," *Sadhana*, vol. 36, no. 5, pp. 837–852, 2011.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[6] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in neural information processing systems*, 2017, pp. 2962–2970.

[7] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[8] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[9] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to human quality tts with transformer," *arXiv preprint arXiv:1809.08895*, 2018.

[10] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[11] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[12] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[13] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4225–4229.

[14] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5621–5625.

[15] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[16] W. Wang, S. Xu, and B. Xu, "First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention," *Interspeech 2016*, pp. 2243–2247, 2016.

[17] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4789–4793.

[18] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[19] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482–7491.

[20] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[21] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7304–7308.

[22] "Chinese standard mandarin speech copus (10000 sentences)," http://www.data-baker.com/open_source.html.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] R. Weide, "The carnegie mellon pronouncing dictionary [cmudict. 0.6]," 2005.