INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Forward-Backward Decoding for Regularizing End-to-End TTS

Yibin Zheng[1,2], *Xi Wang*[3], *Lei He*[3], *Shifeng Pan*[3], *Frank K. Soong*[3], *Zhengqi Wen*[1], *Jianhua Tao*[1,2]

[1]Institute of Automation, Chinese Academy of Science, China
[2]School of Artificial Intelligence, University of Chinese Academy of Science, China
[3]Microsoft China, Beijing, China

yibin.zheng@nlpr.ia.ac.cn, xwang@microsoft.com

Abstract

Neural end-to-end TTS can generate very high-quality synthesized speech, and even close to human recording within similar domain text. However, it performs unsatisfactory when scaling it to challenging test sets. One concern is that the encoder-decoder with attention-based network adopts autoregressive generative sequence model with the limitation of "exposure bias". To address this issue, we propose two novel methods, which learn to predict future by improving agreement between forward and backward decoding sequence. The first one is achieved by introducing divergence regularization terms into model training objective to reduce the mismatch between two directional models, namely L2R and R2L (which generates targets from left-to-right and right-to-left, respectively). While the second one operates on decoder-level and exploits the future information during decoding. In addition, we employ a joint training strategy to allow forward and backward decoding to improve each other in an interactive process. Experimental results show our proposed methods especially the second one (bidirectional decoder regularization), leads a significantly improvement on both robustness and overall naturalness, as outperforming baseline (the revised version of Tacotron2) with a MOS gap of 0.14 in a challenging test, and achieving close to human quality (4.42 vs. 4.49 in MOS) on general test.

Index Terms:Forward-backward, regularization, encoder-decoder with attention, end-to-end, joint-training, TTS

1. Introduction

Recently, with the rapid development of neural network, end-to-end generative text to speech (TTS) models, such as Tacotron and its varieties [1, 2, 3, 4] are proposed to simplify traditional TTS pipeline [5, 6, 7, 8] with a single neural network. The whole text sequence and corresponding frame-level acoustic features could be effectively learned in a unified network, and with the help of WaveNet [9] like neural vocoder, the quality and naturalness of synthesized audios are greatly improved, and even comparable with human recordings within similar domain text. The end-to-end TTS model is based on encoder-decoder framework [10], which has been widely adopted for sequence generation tasks, such as end-to-end speech recognition [11] and neural machine translation (NMT) [12, 13]. This encoder-decoder framework also brings appealing properties, such as little requirement for feature engineering or prior domain knowledge, unified network instead of fragment components, flexible transformation and etc, making it easier to construct a high quality and expressive TTS system [14, 15, 16, 17].

However, in the framework of end-to-end TTS, the decoder is an autoregressive structure which will prevent the usage of global or future output information during training and inference. That is, while generating current frame, one can only leverage the generated frames but not the future frames ungenerated so far. This forward-decoding process will constrain the usage of global or future information. What's worse, a tiny mistake made early could quickly be amplified and propagated alongside the sequence (so-called exposure bias problem [18]). Further, such issue may become severe when the test sequences are not matched with training data, and would become much severer as the sequence length increases. Various attention mechanisms [19, 20, 21] have been incorporated to find a more accurate mapping from encoder to decoder. However, due to the constrain of autoregressive generation, the decoding bias problem could not be well addressed only by attention mechanism itself. In [18], the author provides a simple way to address the "exposure bias", namely scheduled sampling [18], which is done by combing the predicted probability with ground-truth label. To better leverage the global or future information, there are also some attempts in speech recognition [11] and NMT [12, 22], which try to improve the performance by integrating the predicted probability from forward and backward decoding sequences. However, regarding TTS outputs are continuous space, these methods [12, 18, 22] could not be directly employed in TTS.

Actually, the whole utterance is known during training, and with the help of estimation of future information, the generating sequence would be much more robust. Then the accumulated error would be fixed soon and would not be propagated for a long way. The assumption is based on the agreement between forward and backward decoding while generating sequence, and thus the attention should be more accurate. In this paper, we propose two novel methods to realize this forward-backward decoding sequence. 1) Train two directional models, namely L2R (which generates targets from left-to-right) and R2L (which generates targets from right-to-left), and iteratively update the model by a novel training objective with additional consideration of minimizing both directional model divergence at the same time, in which the latter one servers as a measure of exposure bias of the currently evaluated model. 2) With shared encoder, train bi-directional decoders with their own attentions. By adding regularization term for forward-backward decoders to the training objective, the forward hidden states are forced to be well close to the backward ones. Thus, it could encourage the hidden representations of a unidirectional decoder to embed some useful information about the future. For both two methods, a joint training strategy is proposed to make forward and backward decoders improve each other in an interactive update process. Furthermore, the backward network could be omitted during inference, leading to the model that ideally does not introduces any latency and does not add any computation compared to the standard unidirectional end-to-end TTS model.

Samples of synthesized speech are available at https://vancycici.github.io/fbdecode/.

http://dx.doi.org/10.21437/Interspeech.2019-2325

2. Proposed Methods

To better leverage the global or future information as well as to alleviate the exposure bias problem, we describe in depth the two proposed methods that integrate forward and backward decoding sequences here.

2.1. Model regularization by bidirectional agreement

To predict future as well as to deal with the exposure bias problem, we try to maximize the agreement between the generated spectrograms from L2R and R2L end-to-end TTS models, and divide the training objective into two parts: one for the standard L_2 loss of each model, and the other for regularization terms that indicate the divergence of L2R and R2L models based on the current model parameters.

2.1.1. Model regularization

Given a text sequence $x = (x_1, x_2, ..., x_T)$ and its target mel spectrograms $y = (y_1, y_2, ..., y_{T'})$, let $P(y|x; \overrightarrow{\theta})$ and $P(y|x; \overleftarrow{\theta})$ be L2R and R2L model, in which $\overrightarrow{\theta}$ and $\overleftarrow{\theta}$ are corresponding model parameters. Specifically, L2R model can be decomposed as $P(y|x; \overrightarrow{\theta}) = \prod_{t=1}^{T'} P(y_t|y_{<t}, x; \overrightarrow{\theta})$, which means L2R model adopts previous targets $y_1, ..., y_{t-1}$ as history to predict current target y_t at each step t, while R2L model similarly can be decomposed as $P(y|x; \overleftarrow{\theta}) = \prod_{t=T'}^{1} P(y_t|y_{>t}, x; \overleftarrow{\theta})$ and employs future targets $y_{t+1}, ..., y_{T'}$ as history to predict current target y_t at each step t.

Since L2R and R2L models are different chain decompositions of the same output sequence, output sequence of these two models should be ideally identical:

$$P(y|x; \overrightarrow{\theta}) = P(y|x; \overleftarrow{\theta}) \quad (1)$$

However, the above equation will not hold if these two models are optimized separately by standard loss of each model. To satisfy this constrain, we introduce a L_2 regularization term into the training objective. For L2R model, the new training objective now becomes:

$$L(\overrightarrow{\theta}) = \sum_{n=1}^{N} L_2(y_{\overrightarrow{\theta}}^{(n)}, y^{(n)}) + \lambda \sum_{n=1}^{N} L_2(y_{\overrightarrow{\theta}}^{(n)}, y_{\overleftarrow{\theta}}^{(n)}) \quad (2)$$

where λ is a hyper-parameter for regularization term. The regularization term will guide the training process to reduce the disagreement between L2R and R2L model.

Considering the symmetry of L2R and R2L model, L2R model can also serve as the discriminator to punish bad generation candidates generated from R2L. Based on the above, L2R and R2L model can act as helper systems for each other in a joint training process.

Though both L2R and R2L are needed at training time, we could use either L2R or R2L model during inference. This leads to equal amount of computations, compared to standard unidirectional model (at inference time).

2.1.2. Joint training for model regularization of L2R & R2L

To simultaneously optimize these two models, we design a novel training algorithm and the overall training objective is defined as the sum of objectives in both directions:

$$L(\theta) = L(\overrightarrow{\theta}) + L(\overleftarrow{\theta}) \quad (3)$$

As illustrated in Fig. 1, the whole training process contains two major steps: pre-training and joint-training. Firstly, we pre-train both L2R and R2L models with standard loss of each end-to-end TTS model. Next, based on the pre-trained models, we jointly optimize L2R and R2L models with an iterative process. In each iteration, we fix R2L model and use it as an auxiliary helper system to optimize L2R models with Eq.2, and at the same time, we fix L2R model and use it as an auxiliary helper system to optimize R2L model. This training process could be carried out to obtain further improvements because after each iteration both L2R and R2L model are expected to be improved with regularization method.

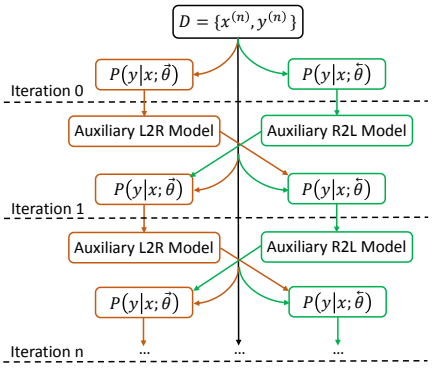

Figure 1: *Illustration of joint training of L2R & R2L model.*

2.2. Bi-directional decoder regularization

The method in Sec. 2.1 is operated in the model level, which is similar to data augmentation (since we have to generate pseudo $< text, audio >$ pairs from reversed model). However, for the data generation is very time consuming and not effective. We consider the second method to further implement forward-backward decoding. Since the unidirectional decoders tend to be more accurate at their early decoding steps, which leads to a strategy that forward and backward decoder can be integrated to boost the final performance. As shown in Fig. 2, in which the forward decoder is trained with a subtask of backward decoding by sharing a single encoder and vice versa, aiming at a regularization effect of optimization for both direction decoder.

2.2.1. Bi-directional decoder regularization

In Fig. 2, the shared encoder represents an input text sequence $x = (x_1, x_2, ..., x_T)$ into hidden representation of $\boldsymbol{h}$, and the forward decoder computes, at each time step t, a new hidden state $\overrightarrow{s}_t$ in the following way:

$$h_t = encoder(h_{t-1}, x_t) \quad (4)$$

$$\overrightarrow{s}_t = decoder_{\overrightarrow{\theta}}(\overrightarrow{s}_{t-1}, y_{t-1}, \overrightarrow{c}_t) \quad (5)$$

where c_t is the context vector calculated by a location-sensitive attention mechanism [2].

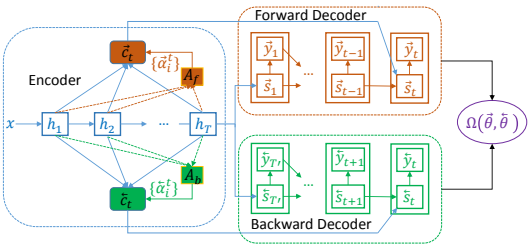

Figure 2: *Bi-direction decoder regularization. Blue, orange, and green parts indicate encoder, forward-decoder and backward-decoder, respectively.*

The forward states summarize the information about current and past elements of the sequence. Similarly, it is very convenient to also process the sequence in the reversed time order, and compute backward states $\overleftarrow{s}_t$ similarly to Eq.5:

$$\overleftarrow{s}_t = decoder_{\overleftarrow{\theta}}(\overleftarrow{s}_{t+1}, y_{t+1}, \overleftarrow{c}_t) \quad (6)$$

The backward states summarize the information about current and future elements of the sequence. To encourage the hidden states of a unidirectional decoder to embed some useful information about the future, we add a regularization term, as highlighted in Fig. 2. The idea is to penalize forward hidden states $\overrightarrow{s}_t$ that are distant from the backward ones $\overleftarrow{s}_t$. With this regard, one can add a regularization term that encourage the network to minimize the L_2 distance between forward and backward hidden states:

$$\Omega = \frac{1}{T'}\sum_{t=1}^{T'} \|\overrightarrow{s}_t - \overleftarrow{s}_t\|^2 \quad (7)$$

The total objective to be minimized thus becomes a weighted sum of the standard loss plus the regularization term:

$$\tilde{L}(\theta) = L(\overrightarrow{\theta}) + L(\overleftarrow{\theta}) + \lambda\Omega(\overrightarrow{\theta}, \overleftarrow{\theta}) \quad (8)$$

where λ is a hyper-parameter controlling the importance of the penalty term, and $\tilde{L}(\theta)$ is the total loss.

Note the backward states are needed only at training time. During inference, the part of the model computing backward states can be omitted. This leads to the architecture particularly suitable for practical use, since it requires exactly the same amount of computations needed for standard unidirectional decoder (at inference time). Another remarkable aspect of this technique is that Eq.6 is based on the backward states $\overleftarrow{s}_t$, that provide a summary of the full future part of the mel spectrograms sequence. This means that our method could capture not only short-term future dependencies, but also long-term ones.

2.2.2. Joint-training of bi-directional decoder regularization

In practice, it is hard to train the whole network from scratch jointly since the revered decoder cannot provide useful enough information at the beginning of training. To simultaneously optimize bi-directional decoders, we design a novel training algorithm, which is similar to that in Fig. 1. The whole training contains two major steps: pre-training and joint-training. First, we pre-train both forward and backward jointly without adding the regularization term. Next, based on the pre-trained bi-directional decoders, we jointly optimize forward and backward decoders with an iterative process using Eq.7. In each iteration, we fix backward decoder and use it as a helper to optimize forward decoder, and vice versa.

3. Experiments

In this section, we conduct experiments to evaluate our proposed methods a 20-hour, 16kHz, 16bit speech corpus, which is recorded by a professional enUS female speaker. All the subjective tests are evaluated by at least 10 native judges from Microsoft crowdsourcing UHRS (Universal Human Relevance System) platform.

3.1. Model details

For our baseline, we use similar architecture and hyper parameters as Tacotron2 [2] except a few details. We test our proposed methods with both phoneme and character inputs for end-to-end model training. The decoder outputs 80-channel mel spectrograms, one frame at a time. We train a WaveNet vocoder conditioned on mel spectrograms with a constant learning rate of 10^{-4}, and use it for waveform generation. Our WaveNet is a smaller one than [2], with only 12 dilated layers [23].

The L2R and R2L model have the same architecture with the baseline model. For joint training algorithm of model regularization, we find that this method is very time consuming (since we have to generate pseudo $< text, audio >$ pairs from reversed model in each iteration) and thus only 1 full iteration is performed to be evaluated here. Besides, we find that the R2L model gets comparable results with the L2R model, thus only the result of the L2R model is reported.

As for bi-directional decoder regularization, the architectures of encoder and decoder are kept the same with the baseline model. We perform 5 full iterations for the joint training of bidirectional decoder. Besides, we find that the forward decoder gets comparable results with the backward decoder, thus only the result of the forward decoder is reported.

Hyper-parameter λ in Eq.8 is set as 1.0 and all the models are trained with 4 Nvidia Tesla P100 using Adam optimizer [24], with initial rate of 10^{-3} and decays exponentially after 50000 steps.

3.2. Evaluation

3.2.1. Evaluation on out-of-domain text

Firstly, we evaluate the proposed methods on out-of-domain text. In this section, we collect a large challenging test set, including news, long facts, foreign words, difficult abbreviation, numbers, letters, strange text normalization (TN) extension, etc. For subjective evaluation, we randomly select 50 test utterances from the challenging test set. We first conduct two AB preference tests on these 50 audios pairs generated by baseline and our proposed two methods respectively. In this AB preference test, we use character as the input for the end-to-end TTS training. The AB preference test results are shown in Fig. 3, in which **Baseline** denotes the end-to-end baseline model (revised version of Tacotron2 [2]), **Bi-L2R** denotes the joint-trained L2R model introduced in Sec. 2.1, and **Bi-Forward-Decoder** represents the joint-trained forward decoder by adding hidden state regularization that introduced in Sec. 2.2. We find that both two proposed methods consistently receive more preferences than the baseline model. These results confirm that the proposed two regularization methods by introducing agreement between forward and backward decoding sequence can help handling exposure bias problem and improving the synthesized audios' naturalness. Specifically, instead of combining backward model during inference, both two proposed methods utilize the intrinsic connection between forward and backward decoding models to guide the learning process. The forward and backward decoding models are expected to adjust in disagreement cases and then the exposure bias problem of them can be alleviated. And among these models, the **Bi-Forward-Decoder** gains the most preferences (60%). It shows from judges' comments that **Bi-Forward-Decoder** tends to generate more natural and human-like speech, meanwhile it reduces the probability of running into pronunciation difficulties, e.g., when handling names or out of vocabulary (OOV) words. Therefore, model **Bi-Forward-Decoder** is selected for further more experiments.

We further train models **Bi-Forward-Decoder** and **Baseline** with phoneme as input. Together with character-based models, 5-point mean opinion score (MOS) tests are conducted. Tab. 1 shows the out-of-domain evaluation results of different models. It's noticed that the proposed model **Bi-Forward-Decoder** performs better than baseline model **Baseline** no mat-

ter what kind of input representation is used. And a much more obvious advantages of our proposed method could be observed when using character as input. This could be explained by that, compared with phoneme-based models, character-based models often suffer from pronunciation issues caused by grapheme-to-phoneme, and our proposed method is helpful to correct such issues since it could use the global information of an utterance to give more proper pronunciation. And among these systems, the **Bi-Forward-Decoder** (with phoneme as input) obtains the best performance with a MOS of 4.26.

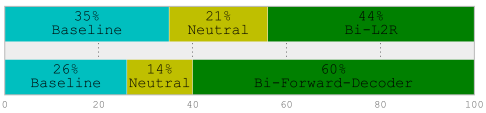

Figure 3: *The results of AB preference test by using character as input, with confidence level of 95% and p-value* < 0.0001.

Table 1: *The MOS of different models in out-of-domain test, with confidence level of 95%.*

Inputs	Character	Phoneme
Baseline	3.77	4.12
Bi-Forward-Decoder	**4.07**	**4.26**
Recording	4.48	4.48

Secondly, we select another 100 test utterances from the large challenge test set for intelligibility test. This set is much more challenging with many abbreviation, letters, numbers, repeated words, OOV, long sentences, etc, which may lead the attention collapses. The number of words in these 100 test utterances is 2261. Each test utterance will be evaluated by two native judges, and both unintelligible and unnatural word will be marked. In this task, we count the following two matrices and the results are shown in Tab. 2 and Tab. 3. These two matrices include: case level intelligible rate which means the proportion of the cases without any words marked as "unintelligible" in a test case; case level natural rate which means the proportion of the cases without any word marked as "unintelligible" or "unnatural" (such as emphasis on the wrong syllables or words, or unnatural pitch) in a test case. It's observed that both the proposed character-based and phoneme-based **Bi-Forward-Decoder** significantly improve the model's intelligibility and naturalness, with an absolute improvement of 4.0% and 4.5%, respectively for character-based model, 1.6% and 1.8% , respectively for phoneme-based model (which is not that significant as character-based model, but still beneficial).

Apart from the quantitative analysis, we also give a case study to better understand how the regularization method works. We show the alignments of the joint-trained **Bi-Forward-Decoder** and **Bi-Backward-Decoder** with the same text, as shown in Fig. 4. As we can see, the alignment of forward and backward decoder is almost the same after regularization, which means the forward and backward decoder reach a high degree of agreement.

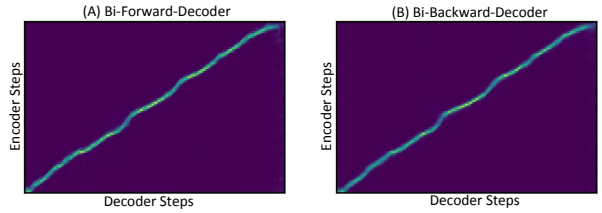

Figure 4: *Attention alignments on a test utterance.*

Table 2: *Case level intelligible rate and natural rate of character-based models.*

Models	Baseline	**Bi-Forward-Decoder**
case level intelligible rate	95.5%	**99.5%**
case level natural rate	93.5%	**98.0%**

Table 3: *Case level intelligible rate and natural rate of phoneme-based models.*

Models	Baseline	**Bi-Forward-Decoder**
case level intelligible rate	96.8%	**98.4%**
case level natural rate	95.6%	**97.8%**

3.2.2. Evaluation on in-domain text

Apart from the out-of-domain evaluation, we also evaluate the proposed method on relative in-domain text, like appropriate text length and content. In this section, only the model **Bi-Forward-Decoder** (which performs best on out-of-domain text) is included for comparison. We randomly select 50 test utterances (not included in training set) from our internal dataset. Tab. 4 shows the in-domain evaluation results of different models. We find that the both the proposed character-based and phoneme-based **Bi-Forward-Decoder** perform better than the baseline model **Baseline**, with a gap of 0.14 and 0.06 in MOS, respectively. This trend is similar as that on out-of-domain evaluation, which further confirms the effectiveness of the proposed regularization method. Meanwhile, it's noticed that phoneme-based model **Bi-Forward-Decoder** obtains the best performance with a MOS of 4.42, which is quite close to recording (4.49). Examination of judges' comments also show that **Bi-Forward-Decoder** performs much better on overall prosody, sounds more expressive, stable and clearer than **Baseline**.

Table 4: *The MOS of different models in in-domain test, with confidence level of 95%.*

Inputs	Character	Phoneme
Baseline	4.10	4.36
Bi-Forward-Decoder	**4.24**	**4.42**
Recording	4.49	4.49

4. Conclusions

In this paper, we propose two efficient regularization training approaches to the end-to-end TTS framework, aiming to improve the robustness of the model. Relying on the optimization of the agreement between forward and backward decoding sequence, the forward decoder could be better optimized with both global and future information of the output, thus gains much more stable, expressive results but without any additional computation during inference. Experimental results demonstrate that our methods (especially the bidirectional decoder regularization method), achieves a significant improvement on robustness and naturalness on both in-domain and out-of-domain evaluation. However, due to the model characteristic of encoder-decoder based framework, it still suffers from incorrect issues on a few extremely rare cases. We will keep on this work to improve the stableness of end-to-end TTS .

5. Acknowledgements

The author would like to thank Shujie Liu and Fei Tian from Microsoft research with fruitful discussion.

6. References

[1] 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," in *INTERSPEECH 2017, Conference of the International Speech Communication Association, Makuhari, Stockholm, Sweden, August*, 2017, pp. 4006–4010.

[2] 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.

[3] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," *arXiv preprint arXiv:1808.10128*, 2018.

[4] Y. Wang, R. Skerry-Ryan, Y. Xiao, D. Stanton, J. Shor, E. Battenberg, R. Clark, and R. A. Saurous, "Uncovering latent style factors for expressive speech synthesis," *arXiv preprint arXiv:1711.00520*, 2017.

[5] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[6] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," in *Eurospeech, Rhodes, Greece, 1997. Conference Proceedings*, 1997, pp. 601–604 vol. 1.

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

[8] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966.

[9] A. v. d. 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," *arXiv preprint arXiv:1609.03499*, 2016.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] S. S. Masato Mimura and T. Kawahara, "Forward-backward attention decoder," in *INTERSPEECH 2018, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September*, 2018, pp. 2232–2236.

[12] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li *et al.*, "Achieving human parity on automatic Chinese to English news translation," *arXiv preprint arXiv:1803.05567*, 2018.

[13] Y. Xia, F. Tian, L. Wu, J. Lin, T. Qin, N. Yu, and T.-Y. Liu, "Deliberation networks: Sequence generation beyond one-pass decoding," in *Advances in Neural Information Processing Systems*, 2017, pp. 1784–1794.

[14] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[15] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[16] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," *arXiv preprint arXiv:1808.01410*, 2018.

[17] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 4485–4495.

[18] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[19] 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.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[21] 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.

[22] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Agreement on target-bidirectional neural machine translation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 411–416.

[23] S. Arik, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-speaker neural text-to-speech," *arXiv preprint arXiv:1705.08947*, 2017.

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