RobuTrans: A Robust Transformer-Based Text-to-Speech Model

Naihan Li,[*1,4,5] **Yanqing Liu,**[2] **Yu Wu,**[3] **Shujie Liu,**[3] **Sheng Zhao,**[2] **Ming Liu**[1,4,5]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[2]Microsoft STC Asia
[3]Microsoft Research Asia
[4]CETC Big Data Research Institute Co.,Ltd, Guiyang
[5]Big Data Application on Improving Government Governance CapabilitiesNational Engineering Laboratory, Guiyang
lnhzsbls1994@163.com
{yanqliu, Wu.Yu, shujliu, szhao}@microsoft.com
csmliu@uestc.edu.cn

Abstract

Recently, neural network based speech synthesis has achieved outstanding results, by which the synthesized audios are of excellent quality and naturalness. However, current neural TTS models suffer from the robustness issue, which results in abnormal audios (bad cases) especially for unusual text (unseen context). To build a neural model which can synthesize both natural and stable audios, in this paper, we make a deep analysis of why the previous neural TTS models are not robust, based on which we propose RobuTrans (Robust Transformer), a robust neural TTS model based on Transformer. Comparing to TransformerTTS, our model first converts input texts to linguistic features, including phonemic features and prosodic features, then feed them to the encoder. In the decoder, the encoder-decoder attention is replaced with a duration-based hard attention mechanism, and the causal self-attention is replaced with a ”pseudo non-causal attention” mechanism to model the holistic information of the input. Besides, the position embedding is replaced with a 1-D CNN, since it constrains the maximum length of synthesized audio. With these modifications, our model not only fix the robustness problem, but also achieves on parity MOS (4.36) with TransformerTTS (4.37) and Tacotron2 (4.37) on our general set.

1 Introduction

Speech synthesis (also known as text to speech, TTS) has a pivotal role in a wide range of speech-related applications. Owing to the development of deep learning techniques, modern TTS pipelines make a step from HMM-based statistical parametric TTS models (Maia, Zen, and Gales 2010), neural acoustic models (Ze, Senior, and Schuster 2013) to neural end-to-end TTS models (Wang et al. 2017; Shen et al. 2017; Li et al. 2018).

Along with the success of neural machine translation, neural sequence to sequence models are applied to TTS tasks, such as Tacotron2 (Shen et al. 2017) and TransformerTTS (Li et al. 2018). The neural sequence to sequence model usually contains three components: an encoder, a decoder and an attention mechanism between them. The encoder is used to convert the input text into a semantic space, based on which the decoder generates the spectrums, with the guidance of the attention mechanism to decide when to pronounce which word. Based on the generated spectrums, a vocoder (Van Den Oord et al. 2016) is leveraged to synthesizes the final audios.

By this method, the synthesized audio has excellent naturalness on general input texts like those in the training set, some even achieve close-to-human quality (Li et al. 2018). However, when the input texts are unusual[1], abnormal spectrums are generated, leading to bad audios. We summarize and categorize these bad audios into following major types: 1) some words may be unclear even missed, or on the contrary, duplicated; 2) the decoding procedure stops too early or too late; 3) the decoding procedure can’t stop until the pre-defined maximum length is reached. The vulnerability and instability of these models limit their applicability to a broader range of tasks, which require robust performance on diverse inputs, such as voice assistant and vehicle navigator.

We conduct a detailed study on the above bad audios generated by TransformerTTS, finding that abnormal audios always appear with disordered attention alignments. To deal with such a problem and ensure the monotonic correspondence from the phoneme sequence to the acoustic sequence, Zhang, Ling, and Dai (2018) propose a forward attention method for higher attention stability, in which the attention probabilities at each time step are computed recursively using a forward algorithm. Raffel et al. (2017) propose a forced monotonic attention mechanism, where at each output time step, the decoder inspects memory entries in left-to-right manner starting from where it left off at the previous output time step and chooses a single one to attend to.

We test them and find none of them can completely get rid of abnormal cases, instead gives rise to other issues such as higher speech rate and weird rhythm.

In this paper, we remove the encoder-decoder attention and apply a duration-based hard attention to copy encoder hidden states to their corresponding frames, forcing the de-

[*] Work done during internship at Microsoft STC Asia.

[1] Such as URL, a sequence of numbers, and other texts which are out of the domain of the training data

coder to generate correct content. To have a holistic view of the whole input as the original attention mechanism, we replace the causal self-attention layer in the decoder with a pseudo non-causal attention (PNCA) to not only consider previously decoded results, but also attend to subsequential contexts. To improve the audio naturalness for long input, we remove the position embedding and rely on a 1-D CNN to model the relative position information. Furthermore, instead of using the text, we leverage linguistic features (including phonemic and prosodic features) as the input of the encoder, of which the prosodic features have a great contribution to the prosody of results. With the above adaptations, our model manages to thoroughly eliminate abnormal results, meanwhile achieve the on parity naturalness with previous neural TTS models. We conduct experiments on two test sets, including a general set and a bad-case set. Our model doesn't make any mistake for the samples in the bad-case set, at the same time, it achieves on parity MOS (4.36) with TransformerTTS (4.37) and to Tacotron2 (4.37) on the general set.

In summary, our contributions can be listed as follows: 1) We conduct a detailed study to show why previous neural TTS models are not robust. 2) We employ a duration-based hard attention to effectively improve the robustness of our model, meanwhile propose a pseudo non-causal attention, which significantly contributes to the naturalness of synthesized audio by providing a holistic view of the input sequence for each decoding step.

Besides, the ability to synthesize long sequences is improved by dispensing the position embedding and relying on 1-D CNN instead, and the naturalness of synthesized audio is further enhanced by leveraging linguistic features.

3) Our model achieves comparable performance on quality and naturalness to previous neural TTS models, meanwhile shows excellent robustness for various input patterns.

2 Why Previous Neural Models Unstable

In this section, we first briefly introduce TransformerTTS (Li et al. 2018), the state-of-the-art Neural TTS model; base on this model, we make a deep analysis on three factors which makes it unstable.

TransformerTTS (Li et al. 2018) is a neural TTS model which combines Transformer (Vaswani et al. 2017) and Tacotron2 (Shen et al. 2017). As shown in Figure 1, given the input text, a text-to-phoneme converter is first used to get the phoneme sequence. With a CNN as the encoder pre-net, context features are extracted to be the input of the encoder. The mel spectrum frames are also processed by a 2-layer fully connected network with $relu$ activation. Position information is injected by adding two position embeddings to the output of the encoder and decoder pre-nets respectively. The encoder is built with stacks of several identity blocks, each contains two sub-networks: a self-attention and a feed forward network. The decoder has the similar structure, while the self-attention is causal to attend to only the previously decoded frames, and an extra encoder-decoder attention is leveraged to attend to encoder hidden states.

Based on the final hidden states of the decoder, mel spectrum frames are generated autogressively with a linear layer followed by a post-net, which stops when a stop token is predicted by a separate linear projection.

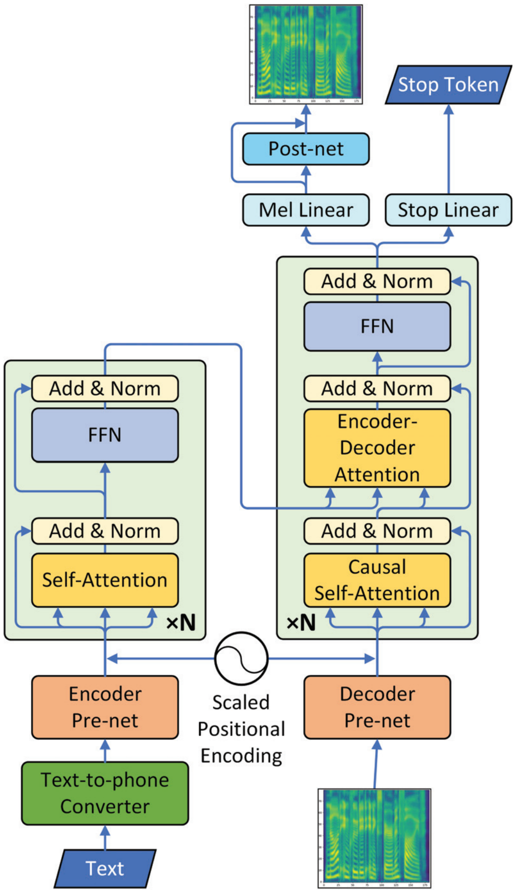


Figure 1: Architecture of TransfomerTTS.

Similar to Tacotron2, TransformerTTS also borrows techniques from neural machine translation (NMT) community. Some designs for NMT, however, do not fit TTS tasks, which is the root cause of the instability and robustness problem. In the following section, we summarize three main drawbacks.

2.1 Unconstrained Encoder-decoder Attention

TransformerTTS borrows the soft attention mechanism from NMT, which enables the decoder to attend to arbitrary parts at the source side for each step. The mechanism is reasonable for NMT because the word order of two languages may be different, which means the last word on the target side may correspond to the first word on the source side. In contrast, text to speech has a unique property, which is **monotonous continuous correspondence**, meaning that if the i-th step attends to the j-th word at the source side, the $(i+1)$-th step must attend to the $(j+n)$-th word $(1 \geq n \geq 0)$, as shown in the left picture in Figure 2.

Previous models ignore this constraint, and learn the alignment from the data totally, resulting in incorrect alignments for special inputs. The right picture in Figure 2 shows an example of an abnormal alignment. On the one hand, the attention mechanism skips the second word and attends to

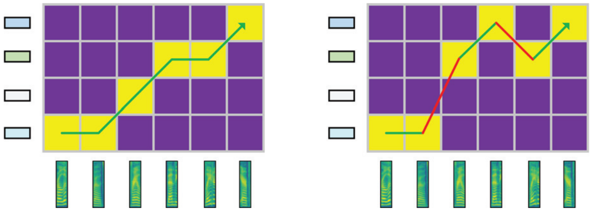


Figure 2: Normal and abnormal alignments of encoder-decoder attention. Mel spectrum frames (queries) are ranged horizontally, while encoder hidden states (keys) are vertical. Left: normal alignment; the focus along keys are continuous and monotonous. Right: Abnormal alignment; the red line represents the skipping as well as retreating advance.

the third word at the third step, while on the other hand, after attending to the forth word at the fourth step, it attends to the third word again. These two cases definitely output bad cases. Although some variation of attention mechanisms (e.g. forward attention) has tried to construct a monotonous continuous correspondence between encoder and decoder, they cannot completely eliminate bad cases. Details will be shown in Section 4.6.

2.2 Imprecise Stop Prediction

Different from NMT models which predict one token each time until a stop token is predicted, TTS models predict mel vectors and require a separate classifier to decide when to stop. However, this stop predictor is usually unreliable due to two reasons: 1) Each case of TTS consists of hundreds of decoding steps but only one stop step, leading to an imbalance continue/stop classification problem and an undesirable performance on stop prediction. Although imposing a larger weight on the ”stop” class can efficiently relieve this issue, the stop prediction still makes mistakes for some specific inputs. For example, if the input text is twenty consecutive ”0”s (”00 . . . 0”), the generated audio is likely to contain more or less than twenty. 2) The stop token is autoregressively predicted only conditioning on the hidden state of each step, which contains no explicit information of whether the input is all and once pronounced.

2.3 Unseen position embedding

Transformer injects the position information by adding position embeddings to the inputs of its encoder and decoder:

$$PE(pos, 2i) = \sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \tag{1}$$

$$PE(pos, 2i+1) = \cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \tag{2}$$

However, when the input sequence is longer than the normal length as it is in the training set, the position indexes can be extremely large, of which the corresponding position embeddings are unseen for both the encoder and decoder. Therefore, the input sequences to which these position embeddings are added may confuse the model, leading to the occurrences of abnormal outputs.

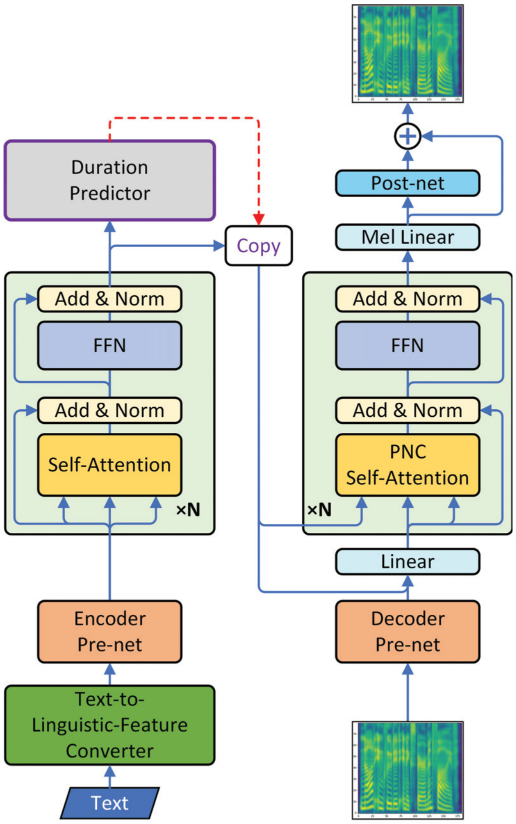


Figure 3: Architecture of RobuTrans.

3 Robust Transformer-based TTS

In this section, we will introduce RobuTrans, a robust neural TTS model based on Transformer, as shown in Figure 3. The main processing pipeline is, on the one hand, the input text is firstly converted into the sequences of linguistic features, then through Encoder Pre-net (a 3-layer CNN) and Encoder. Encoder hidden states are fed into Duration Predictor, which predicts their durations, then they are tiled into frame-level. On the other hand, the shifted mel spectrum is firstly processed by Decoder Pre-net, which is a two-layer fully connected network with $relu$ activation. The tiled encoder hidden states and processed mel spectrum frames are concatenated and then through a linear projection to be fused and have the appropriate dimension for Decoder. The Decoder hidden states are processed by a linear projection to obtain the decoded spectrum, then through Post-net (a 5-layer CNN) to obtain the final spectrum.

RobuTrans differs from TransformerTTS in following aspects: 1) The input of Encoder is linguistic features, which consists of phonemic and prosodic features; 2) The position embedding in the Encoder and Decoder is removed; 3) The encoder-decoder attention is replaced with a duration based hard attention; 4) The causal self-attention in Decoder is replaced with pseudo non-causal attention.

3.1 Text-to-Linguistic-Feature Converter

We first convert the input text into linguistic features, which consist of phonemic and prosodic features and then consumed by Encoder. To obtain the phonemic features, a rule-based system is used for the grapheme-to-phoneme conversion, which generates the phonemic categorical features[2]. The prosodic feature includes tone and break index, which are predicted by a conditional random field (CRF) model with syntactic and contextual information as in Qian et al. (2010). The prosodic feature plays a critical role in RobuTrans for synthesizing expressive speech, while, on the contrary, harms the quality of TransformerTTS. Two ablation studies will be demonstrated in Section 4.8.

3.2 Duration Predictor

We adopt the structure of the duration predictor as in FastSpeech (Ren et al. 2019), where there are two convolutional layers (kernel size is 3, hidden size is 256) with layer normalization and dropout, together with a linear projection to predict the logarithmic duration of each encoder hidden state. Mean squared error (MSE) is employed as the loss function. To generate the ground truth duration for the model training, speech recognition tools are employed to make the forced alignment between the audio and the phoneme sequence. With the predicted duration, the phoneme-level features are copied and expanded to frame-level features accordingly, as illustrated in Figure 4.

3.3 Pseudo Non-causal Attention

As discussed in Section 2.1, the encoder-decoder attention mechanism is a crucial factor for the instability. However, simply removing this attention will also discard the advantages it brings to the TTS model. The advantages can be considered as the following two aspects. On the one hand, the encoder-decoder attention provides a holistic view of input sequence for the decoder, while on the other hand, it composes frame-level context vectors according to decoder inputs (which are mel frames). These two advantages make great contribution to the decoding procedure, and we propose ”pseudo non-causal attention” (PNCA) to replace the causal self-attention layers as shown in Figure 4, which not only inherits the two features above, but also makes the decoding procedure robust.

Let T be the total length of mel spectrum to be decoded, x_i^l be the autoregressive output of step i and layer l, h_i be the tiled encoder hidden state of step i. For the time step t, the PNCA of layer l takes $\left[x_1^{l-1}, x_2^{l-1}, ..., x_t^{l-1}\right]$ [3] and $[h_t, h_{t+1}...h_T]$ as input. Specifically, let $\text{Attention}(Q, K)$ be the multi-head attention,

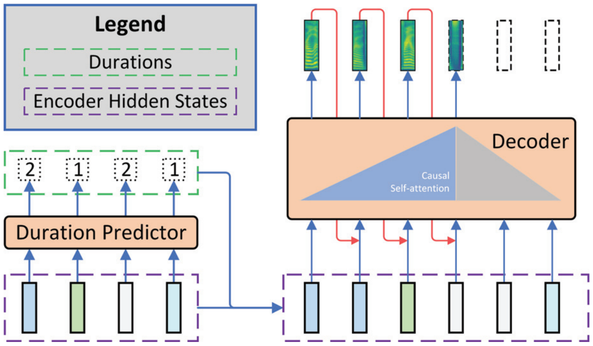


Figure 4: Phoneme-level to frame-level conversion and pseudo non-causal attention (PNCA). The left part of PNCA is causal self-attention, which takes the encoder hidden states fused with padded mel spectrum frames by a linear projection as input, while the right part consumes only the encoder hidden states.

$$y_t^l = \text{PNCA}(x_t^{l-1}, X_{1...t}^{l-1}, H_{t...T}) \tag{3}$$

$$= \text{Attention}(x_t^{l-1}, X_{1...t}^{l-1}) + \text{Attention}(x_t^{l-1}, H_{t...T}) \tag{4}$$

Then y_t^l is added to x_t^{l-1} and consumed by FFN and following residual connection to obtain x_t^l.

By using pseudo non-causal attention, RobuTrans enjoys the following advantages comparing to TransformerTTS: 1) Decoder gets a holistic view of the input sequence, as well as frame-level context vectors, thus the two benefits of vanilla encoder-decoder attention is kept; 2) the output length is determined by the sum of durations of phonemes, thus the stop predictor is removable and the issue in Section 2.2 is addressed; 3) copying encoder hidden states according to their durations explicitly builds a monotonic continuous correspondence between encoder and decoder steps, which provides the instruction for each decoding step and helps deal with the abnormal alignment problem in Section 2.1. With these three advantages, RobuTrans becomes robust and manages to synthesize stable audios, meanwhile the audio quality has no regression.

3.4 Removing Position Embedding

As it is demonstrated in Section 2.3, we find that the position embedding severely constrains the valid length of synthesized audio in our experiments. Specifically, when a text is much longer than those in the training set, the synthesized audio becomes unclear and its prosody is very strange.

Instead of adding this positional embedding, as investigated in the speech recognition community (Mohamed, Okhonko, and Zettlemoyer 2019), we can simply remove the position embedding and count on the CNN used in Encoder Pre-net to model the relative position information in a fixed window. By this alteration, RobuTrans can synthesize longer sequences than those in the training set.

[2] We group the phonemes into different categories, and the categorical feature indicates which groups the phoneme belongs to.

[3] if $l = 1$, x_i^0 is the fusion of padded mel spectrum frame and encoder hidden of step i. We concatenate the h_i and the $(i-1)$-th mel spectrum frame processed by Decoder Pre-net, then through a linear projection.

4 Experiment

4.1 Baseline Model

There are three baseline models to verify both the naturalness and robustness of RobuTrans respectively, which are TransformerTTS, Tacotron2, and FastSpeech. These three models are all trained with the same dataset introduced in Section 4.2.

4.2 Training Setup

We use 4 Nvidia Tesla P100 to train our model. Since the lengths of training samples vary greatly, a fixed batch size will either run out of memory when the batch size is large, or makes the training procedure inefficient and unstable if the batch is small. Therefore, a dynamic batch size is adapted. Each GPU has a memory of 16GB, which can hold 6000 frames (total length of 10∼40 samples), and thus the batch size is 40∼160. For the training set, we use an internal US English dataset, which contains 20 hours of speech from a single professional speaker. 80-channel mel scaled spectrum is extracted from 16k normalized wave, and all the training texts are also normalized. The time consuming for a single training step is 0.55 seconds, and it takes 150,000 steps (about 23 hours) to converge.

4.3 Test Setup

RobuTrans aims to be not only natural but also robust for all input text. Therefore, we respectively test our model with two different test sets for these two aspects.

Robustness test: To test the model robustness, we have a bad-case set consists of 327 sentences, which covers the main categories. We collect them from a large corpus consisting of tens of thousands of sentences. A Tacotron2 model is employed to generate all the audios, of which the corresponding phoneme sequences are then recognized by a speech recognition tool. We calculate the WER of these phoneme sequences with ground truths. Those texts with the highest error rate are collected as our bad case set, which has no overlap with our training set. Some typical error-prone examples are listed in Table 1. The results of this test consists of ”has” and ”doesn’t have” bad cases.

Naturalness test: A 81-sentence test set is randomly selected from the general domain of a large internal corpus (containing millions of sentences). Sentences in these set are similar as training set but have no overlap. To evaluate the naturalness, we conduct both a MOS test (among RobuTrans and baseline models) and CMOS[4] tests (between RobuTrans and baseline models respectively).

All tests are conducted on a crowd-sourcing platform, where the testers are registered by themselves. We didn’t specify anything except for the maximum number of audios (40 sentences for MOS, 30 sentence pairs for CMOS) each tester could listen to in one test, and the tester number (12 for MOS, 9 for CMOS) each sample is listened by.

4.4 WaveNet Vocoder

To obtain audios with high quality, we employ an autoregressive WaveNet vocoder to synthesize the audio with mel sequence as input for all the models, which is trained separately conditioning on ground truth mel spectrums extracted by the audios. The sample rate of ground truth audios is 16000 and frame rate (frames per second) of ground truth mel spectrums is 80. Our autoregressive WaveNet contains 2 QRNN layers and 20 dilated layers, and the sizes of all residual channels and dilation channels are 256. Each frame of QRNN’s final output is copied 200 times to build the conditional input of the 20 dilated layers to fit the same length with audio samples.

4.5 Result

Generated audio samples are accessible in the supplementary materials, including those from the general set generated by RobuTrans and three baseline models, as well as those by RobuTrans from the bad-case set synthesized by RobuTrans listed in Table 1 .

We find that RobuTrans can not only synthesize unusual sentences like a single letter, number and letter series, but also robust for URLs, command lines, and even for some long and meaningless sentences, which are completely out of the domain of our training set.

Robustness test: On our bad-case set (327 sentences), RobuTrans can always synthesize completely correct audios, including single letter, number and letter series, as well as URLs, command lines, and even for some long and meaningless sentences, which are completely out of the domain of our training set. FastSpeech also generates no bad cases, while TransformerTTS and Tacotron2 have 237 and 35 bad cases respectively. Note that, on the one hand, we can qualitatively conclude that RobuTrans and FastSpeech are two robust TTS models on these input patterns, while TransformerTTS and Tacotron2 are not; meanwhile, TransformerTTS is most likely to generate abnormal results among these models. We think the reason for the poor robustness of Transformer is that its decoder has 6 encoder-decoder attention among its 6 blocks, each including 8 heads; each of them has a chance to make mistakes, which makes the model more error-prone. On the other hand, *these numbers of bad cases cannot be used to quantify the robustness of our model as well as baselines*. Specifically, each pattern includes infinite samples, such as duplicate ”zero” for 20, 50 and 100 times, etc., on which Transformer and Tacotron2 always generates bad cases.

Naturalness test: All models generate correct results on this set. Test results are shown in Table 2, and we have following three conclusions. 1) RobuTrans is on parity with Tacotron2, as MOS 4.36 verse 4.37 and CMOS -0.062. 2) RobuTrans is also on parity with TransformerTTS, as MOS 4.36 verse 4.37 and CMOS -0.051. 3) RobuTrans outperforms FastSpeech, as MOS 4.36 verse 4.31 and CMOS $+0.187$. Acousticly, the audio synthesized by FastSpeech with WaveNet vocoder has background noise and unclear pronounciation, which is more obvious when pairwise listened to in CMOS test.

[4] Comparison mean option score, in which the annotator listens to two audios from different models with the same text each time and evaluates how the latter is better than the former with an integer score ranging in $[-3, 3]$. Since the order of the two audios changes randomly, the tester has no idea about their sources.

Table 1: Categories of error-prone text and corresponding examples

Category	Example
Single Letter	W
Number	zero zero zero zero zero zero zero zero two seven nine eight F three forty zero zero zero zero zero six four two eight zero one eight
Spelling	backslash i n t e r n a l dot e x c h a n g e dot m a n a g e m e n t dot s y s t e m m a n a g e
URL	http://office/c16/specs/Specs2/Forms/All%20Office%20Specs.aspx? RootFolder=/c16/specs/Specs2/FrontPage&View={33888BDC-E0CB-4928-AEB7-26607D28009F}
Command Line	$runtime.windows\Speech_OneCore\Engines\TTS\ar-EG\ArEGDiacModel.Bin
Spelling & Number (long, up to 30 secs)	DUB - OWA - zero one JPN - OWA - zero one RED - OWA - zero one RED - OWA - zero two SIN - OWA - zero one SIN - OWA - zero two SYD - OWA - zero one SYD - OWA - zero two Corporate MSG Servers Server Name Exchange Version OS Version Able to Upgrade to WS2003 ?

Model	MOS	CMOS
Tacotron2	4.37 (0.06)	-0.062 (0.088)
TransformerTTS	4.37 (0.06)	-0.051 (0.079)
FastSpeech	4.31 (0.06)	$+0.187$ (0.085)
RobuTrans	4.36 (0.06)	-
Recording	4.69 (0.06)	-

Table 2: MOS and CMOS test results. In ”MOS” and ”CMOS” column, the number in brackets is confidence interval radius with confidence level 0.95. Note that in the ”CMOS” column, all scores measure the *improvement* of RobuTrans comparing to the three baseline models.

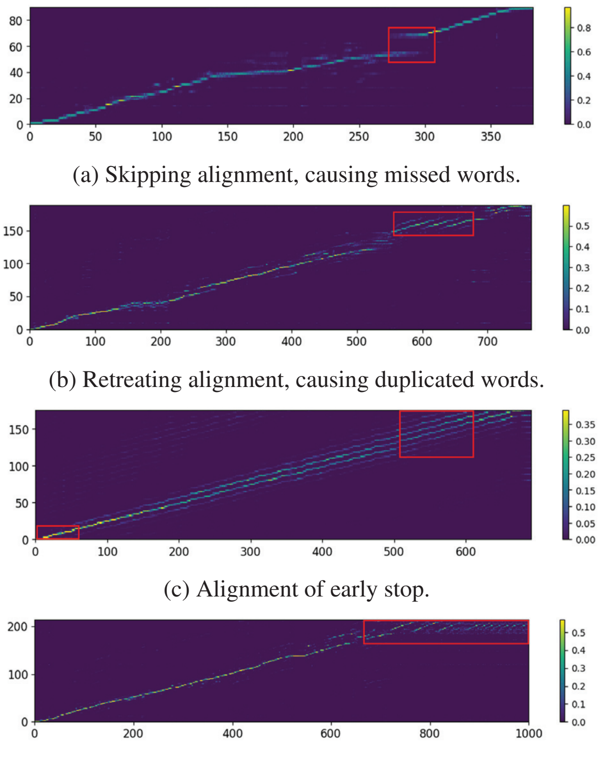
4.6 Other Attention Mechanisms

We conduct experiments investigating the robustness of other attention mechanisms, including forward attention (Zhang, Ling, and Dai 2018), GMM attention (Graves 2013), forced monotonic mechanism (Raffel et al. 2017) and guided attention (Zhu et al. 2019). All these mechanisms generate bad cases, therefore none of them could be part of our robust model. Besides, we find that GMM attention makes the model more stable on long sequences comparing to the vanilla attention mechanism. Forward attention makes inference procedure more unstoppable-prone, forced monotonic mechanism makes the speech rate higher, and guided attention produces audios with weird rhythm.

4.7 Bad Case Analysis

As reported above, RobuTrans can generate completely correct samples in our bad-case set, while both Transformer and Tacotron2 has some errors. To visualize these bad cases, we show alignments from certain heads of certain layers of the TransformerTTS encoder-decoder attention, which are obviously disordered and account for the bad cases.

Missed/duplicated Words: As demonstrated in Section 2.1, there isn't any constraint for the monotonous continuous correspondence of the encoder-decoder alignment, thus some words may be missed or duplicated. The disconnection of the alignment, shown in Figure 5(a), results in the missed words; on the contrary, some words are pronounced for more than once, which can be interpreted by the repeated attention in Figure 5(b).



(a) Skipping alignment, causing missed words.

(b) Retreating alignment, causing duplicated words.

(c) Alignment of early stop.

(d) Alignment of unstoppable prediction.

Figure 5: Alignments of some typical bad cases.

Imprecise Stop: Figure 5(c) shows the alignment of a sample, of which the text is ”zero zero zero ...” (22 repeated ”zero”s). However, there are only 21 ”zero”s are pronounced in the synthesized audio. It can be observed that the alignment becomes confused in the rear part of the decoding procedure (the only one line at the beginning becomes five lines at the end). The reason could be that the decoder has no idea which word it is decoding since all the words are the same, which results in the early stop (similar alignment can also lead to late stop).

Unstoppable Prediction: In Figure 5(d), it can be observed that the alignment keeps repeating the rear words, and cannot stop until the pre-defined maximum decoding length is reached (which is 1000 in this sample).

4.8 Ablation Study

To better understand the impact of the components in RobuTrans, we conduct ablation studies as below.

Pseudo Non-Causal Attention: Pseudo non-causal attention provides a view of subsequential context. We quantify this innovation by an ablation study where the pseudo non-causal attention is changed back to the causal self-attention. The CMOS is -0.291 (CI (95%): 0.077), proving that attending to subsequential frames contributes significantly to the quality of synthesized audio.

Prosodic Feature: To verify the impact of prosodic features, we evaluate the performance without them. We find that the prosody of generated audios becomes weird and unnatural. We conduct a CMOS test and find that removing prosodic features results in a regression with CMOS -0.134 (CI (95%): 0.068), which confirms that the prosodic features play a significant role in RobuTrans.

Linguistic feature for TransformerTTS: When comparing RobuTrans with Transformer, the extra information added by Text-to-Linguistic-Feature Converter may be a factor of an unfair comparison. Therefore, we add the same extra information to TransformerTTS [5], and test its CMOS comparing to the original version. The result is -0.047 (CI (95%): 0.11), which means adding extra information doesn't improve but severely harms the quality (similar result is also obtained on Tacotron2).

5 Related Work

Traditional speech synthesis methods can be categorized into two classes: concatenative systems and parametric systems. Concatenative TTS systems (Hunt and Black 1996) split original waves into small units, and stitch them by some algorithms such as Viterbi (Viterbi 1967) followed by signal process methods (Charpentier and Stella 1986; Verhelst and Roelands 1993) to generate new waves. Parametric TTS systems (Zen, Tokuda, and Black 2009; Ze, Senior, and Schuster 2013; Tokuda et al. 2013) convert speech waves into spectrograms, as well as acoustic parameters, such as fundamental frequency and duration, which are employed to synthesize new audio results.

Traditional speech synthesis methods require extensive domain expertise and may contain brittle design choices. This may be time-consuming and require a lot of resources for manpower. On the other hand, with the rapid development of neural networks, neural TTS has become the mainstream.

Char2Wav (Sotelo et al. 2017) integrates the front-end and the back-end with a seq2seq (Sutskever, Vinyals, and Le 2014; Bahdanau, Cho, and Bengio 2014) model, predicting acoustic parameters for a following SampleRNN (Mehri et al. 2016). This simplifies the complex traditional pipeline. After that, end-to-end TTS models become the research focus, aiming to directly learn the text-to-audio procedure. In the common pipeline, the text is first converted into the spectrum, a highly-compressed representation of the audio, by acoustic models, such as Tacotron (Wang et al. 2017), Tacotron2 (Shen et al. 2017), TransformerTTS (Li et al. 2018) and ClariNet (Ping, Peng, and Chen 2018), then the spectrum is converted into the audio by a neural vocoder. As for the neural vocoder, WaveNet (Van Den Oord et al. 2016) is a powerful model which can generate high-quality audios. Combine the acoustic model and vocoder, neural TTS achieves extraordinary results and significantly outperforms traditional TTS systems.

Though neural TTS shows promising ability, there are still two barriers preventing them from being widely applied to application especially in industry. On the one hand, the inference cannot be real-time since both the acoustic model and vocoder are autoregressive, which mean the prediction of each time step depends on previous steps, thus the inference is serial, and the acoustic model faces the same situation. To tackle this problem, fast vocoders are firstly proposed, such as Parallel WaveNet (Oord et al. 2017), WaveGlow (Prenger, Valle, and Catanzaro 2019), WaveRNN (Kalchbrenner et al. 2018) and LPCNet (Valin and Skoglund 2019). After the speedup on vocoders, parallelization is then investigated on the acoustic model. FastSpeech (Ren et al. 2019) breaks the autoregressive connection in its decoder, and employs WaveGlow, a parallel vocoder, making the inference completely non-autoregressive and two orders faster.

6 Conclusion

In this paper, we first give a deep analysis of why previous neural TTS models are unstable. Among these reasons, the encoder-decoder attention borrowed from NMT is the most critical factor. TransformerTTS and Tacotron2 always generates bad cases on certain patterns, and TransformerTTS is more error-prone since it employs more such attention mechanisms. Besides, the position embedding also constrains the maximum length of generated audio. Based on this analysis, we propose RobuTrans, a robust neural TTS model based on Transformer, which is not only robust even for unseen context but also capable to synthesize natural speech audios, of which the quality is on parity with TransformerTTS and Tacotron2 on a general test domain. We find that FastSpeech is also robust since it employs similar duration-based hard encoder-decoder attention, while our model outperforms it on audio quality and requires no additional teacher model.

7 Acknowledgments

This work is supported in part by National Science Foundation of China under Grant No.61572113, and Big Data Application on lmproving Government Governance CapabilitiesNational Engineering Laboratory Open Fund Project.

References

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

Charpentier, F., and Stella, M. 1986. Diphone synthesis using an overlap-add technique for speech waveforms concate-

[5] Employ this converter to process the input of TransformerTTS then feed into it.

nation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, volume 11, 2015–2018. IEEE.

Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Hunt, A. J., and Black, A. W. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 1, 373–376. IEEE.

Kalchbrenner, N.; Elsen, E.; Simonyan, K.; Noury, S.; Casagrande, N.; Lockhart, E.; Stimberg, F.; Oord, A. v. d.; Dieleman, S.; and Kavukcuoglu, K. 2018. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*.

Li, N.; Liu, S.; Liu, Y.; Zhao, S.; Liu, M.; and Zhou, M. 2018. Neural speech synthesis with transformer network. *arXiv preprint arXiv:1809.08895*.

Maia, R.; Zen, H.; and Gales, M. J. 2010. Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters. In *Seventh ISCA Workshop on Speech Synthesis*.

Mehri, S.; Kumar, K.; Gulrajani, I.; Kumar, R.; Jain, S.; Sotelo, J.; Courville, A.; and Bengio, Y. 2016. Samplernn: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*.

Mohamed, A.; Okhonko, D.; and Zettlemoyer, L. 2019. Transformers with convolutional context for asr. *arXiv preprint arXiv:1904.11660*.

Oord, A. v. d.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; Kavukcuoglu, K.; Driessche, G. v. d.; Lockhart, E.; Cobo, L. C.; Stimberg, F.; et al. 2017. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*.

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

Prenger, R.; Valle, R.; and Catanzaro, B. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3617–3621. IEEE.

Qian, Y.; Wu, Z.; Ma, X.; and Soong, F. 2010. Automatic prosody prediction and detection with conditional random field (crf) models. In *2010 7th International Symposium on Chinese Spoken Language Processing*, 135–138. IEEE.

Raffel, C.; Luong, M.-T.; Liu, P. J.; Weiss, R. J.; and Eck, D. 2017. Online and linear-time attention by enforcing monotonic alignments. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2837–2846. JMLR. org.

Ren, Y.; Ruan, Y.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; and Liu, T. 2019. Fastspeech: Fast, robust and controllable text to speech. *CoRR* abs/1905.09263.

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerry-Ryan, R.; et al. 2017. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*.

Sotelo, J.; Mehri, S.; Kumar, K.; Santos, J. F.; Kastner, K.; Courville, A.; and Bengio, Y. 2017. Char2wav: End-to-end speech synthesis. *ICLR 2017 workshop*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Tokuda, K.; Nankaku, Y.; Toda, T.; Zen, H.; Yamagishi, J.; and Oura, K. 2013. Speech synthesis based on hidden markov models. *Proceedings of the IEEE* 101(5):1234–1252.

Valin, J.-M., and Skoglund, J. 2019. Lpcnet: Improving neural speech synthesis through linear prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5891–5895. IEEE.

Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. In *SSW*, 125.

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

Verhelst, W., and Roelands, M. 1993. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, 554–557. IEEE.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory* 13(2):260–269.

Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. 2017. Tacotron: A fully end-to-end text-to-speech synthesis model. *arXiv preprint*.

Ze, H.; Senior, A.; and Schuster, M. 2013. Statistical parametric speech synthesis using deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 7962–7966. IEEE.

Zen, H.; Tokuda, K.; and Black, A. W. 2009. Statistical parametric speech synthesis. *Speech Communication* 51(11):1039–1064.

Zhang, J.; Ling, Z.; and Dai, L. 2018. Forward attention in sequence-to-sequence acoustic modelling for speech synthesis. *CoRR* abs/1807.06736.

Zhu, X.; Zhang, Y.; Yang, S.; Xue, L.; and Xie, L. 2019. Pre-alignment guided attention for improving training efficiency and model stability in end-to-end speech synthesis. *IEEE Access* 7:65955–65964.