FluentTTS: Text-dependent Fine-grained Style Control for Multi-style TTS

Changhwan Kim¹, Se-yun Um¹, Hyungchan Yoon¹, Hong-Goo Kang¹

¹Yonsei University, Department of Electrical and Electronic Engineering, Seoul, South Korea
[chkim, syum, hcy71]@dsp.yonsei.ac.kr, hkgang@yonsei.ac.kr

Abstract
In this paper, we propose a method to flexibly control the local prosodic variation of a neural text-to-speech (TTS) model. To provide expressiveness for synthesized speech, conventional TTS models utilize utterance-wise global style embeddings that are obtained by compressing frame-level embeddings along the time axis. However, since utterance-wise global features do not contain sufficient information to represent the characteristics of word-level local features, they are not appropriate for direct use on controlling prosody at a fine scale. In multi-style TTS models, it is very important to have the capability to control local prosody because it plays a key role in finding the most appropriate text-to-speech pair among many one-to-many mapping candidates. To explicitly present local prosodic characteristics to the contextual information of the corresponding input text, we propose a module to predict the fundamental frequency (F0) of each text by conditioning on the utterance-wise global style embedding. We also estimate multi-style embeddings using a multi-style encoder, which takes as inputs both a global utterance-wise embedding and a local F0 embedding. Our multi-style embedding enhances the naturalness and expressiveness of synthesized speech and is able to control prosody styles at the word-level or phoneme-level.

Index Terms: emotional speech synthesis, multi-style TTS, style disentanglement, fine-grained style control

1. Introduction
Research on end-to-end neural TTS models has progressed remarkably over the past several years [1, 2, 3, 4]. Most neural TTS models predict acoustic features such as mel-spectrograms or latent space embeddings from input phonetic features, after which the acoustic features are used to synthesize speech waveforms using a neural vocoder [5, 6, 7]. To generate human-like speech signals, neural TTS systems also need to have the capability to support multiple speaker identity characteristics and control the degree of expressiveness with prosody-related information such as speaking speed, emotion, or intonation. However, it is not easy to incorporate these capabilities with conventional TTS models. This is because there are a variety of style options for the output that must be determined from only input text; i.e., it is a one-to-many mapping problem.

One simple but effective method to solve this problem is to utilize additional conditional embeddings that contain the target speaker identity and prosody-related information. In the multi-speaker TTS setting, speaker identity information can be provided in the form of a one-hot vector or a speaker embedding [8, 9, 10]. To provide prosodic variations, previous approaches predicted the prosodic information of an entire input utterance by referencing the global information obtained from processing acoustic features along the time axis [11, 12, 13, 14, 15, 16, 17]. The extracted conditional embedding, called a global style embedding, is added to text embeddings and then used as input text-related information during the decoding stage.

However, in this setting, since the same global style embedding is applied to each text symbol, the decoder has to rely on only the text embedding to represent local prosodic variations. In other words, the global style embedding that averages the characteristics of overall prosodic features cannot provide local prosodic information to the decoder explicitly. In multi-style TTS systems that have multi-speaker and multi-emotion functionalities, prosodic styles can vary rapidly depending on speaker identity and emotion types. Therefore, it is very difficult to find the appropriate pairs or time alignment between text symbols and acoustic frames. Some recent work proposed fine-grained local style control approaches such as estimating prosody information by applying attention between text and global style embeddings [18], learned ranking function [15], or predicting the pitch of input text symbols [19].

Inspired by the approaches described above, we propose a local feature control method to generate more fine-grained style embeddings, which improves local prosodic variations in TTS. Among many types of acoustic features, we use frame-level fundamental frequency (F0) for representing local style. To minimize the prediction complexity of F0 values for various styles, for the utterances of each emotion type, we normalize the F0 values of each speaker by their mean and standard deviation to have a standard normal distribution. Since the number of F0 values in the input utterance is same as that of spectral acoustic feature frames, we use an internal aligner [20] to synchronize timing information with the corresponding text symbols.

During training, our model generates F0 embeddings from phoneme-level reference F0 values using the internal aligner, which are then concatenated with global style embeddings. At the same time, it learns to predict F0 values for input text symbols using global style embeddings provided through conditional layer normalization [21]. During inference, our model generates F0 embeddings from the predicted F0 values and uses them to control local prosodic variations in the generated speech. Our proposed model demonstrates higher mean opinion score (MOS) test scores for naturalness and emotional expressiveness compared to a baseline. Since our model has a module to predict F0 values corresponding to input text symbols, it is possible to explicitly modify the prosody of synthesized signals even in word-level and phoneme-level.

The contributions of our paper are as follows: 1) we propose a neural multi-style TTS model that generates content-dependent fine-grained style features; 2) our model is able to explicitly control local prosodic variations even when the prosody is rapidly changing, such as in emotional speech; 3) our model outperforms a baseline in terms of naturalness, stable alignment, and local prosodic variation.

The organization of our paper is as follows. In Section 2, we introduce the conventional algorithms that are related to our proposed model. In Section 3, we explain the architecture of our model in greater detail. Experimental results and discussion are described in Section 4. The conclusion follows in Section 5.
2. Related Work

Transformer-TTS [3] presented a TTS framework using transformer blocks that consist of multi-head attention structure. Since the multi-head attention can be computed in parallel, Transformer-TTS can be trained much faster than conventional RNN-based models. To have a multi-speaker capability with Transformer-TTS, Multispeech [22] added several modules to improve the accuracy of timing instant between text and speech. It used layer normalization on text embedding before adding positional embedding for preserving positional information. Likewise, it used diagonal constraint [2] on the encoder-decoder attention module for stable convergence. Thanks to the modules introduced earlier, Multispeech was able to alleviate the one-to-many mapping problem in the multi-speaker scenario.

In multi-style scenario, it is essential to provide not only speaker identities but also prosody-related information to the TTS model. Skerry-Ryan et al. [11] presented a reference encoder that consists of the stacking of 2-d convolutional layers to capture local context and a GRU to obtain representative prosodic features at the final hidden state. To control prosody at a fine scale rather than only global scale, several studies represented methods for controlling local prosody. Lei et al. [15] proposed a pretrained ranking function on the Tacotron-based architecture, which estimates the attribute of emotional strength compared with neutral speech. Based on Fastspeech2, Fastpitch [19] proposed an explicit F0 control method using a pitch predictor. The pitch predictor predicts the average pitch value of text by targeting the ground-truth one extracted from an external aligner.

Meanwhile, since the performance of TTS is significantly improved, the next challenge is implementing state-of-the-art TTS model without using external modules. Badlani et al. [20] proposed an internal aligner that learns probabilistic distribution between text and speech using soft alignment, and tried to find the most probable path to make the hard alignment have same characteristic with the soft alignment. The hard alignment contains only binary values (0 or 1), thus it can be used for upsampling the input text embedding to have the length of acoustic feature frames.

In this paper, we implement a multi-style TTS model having a capability of controlling content-dependent fine-grained prosody without using external modules. To generate F0 values for each text symbol, we utilize the hard alignment output from the internal aligner. In addition, we predict the F0 value of each text symbol using the global style information injected by conditional layer normalization [21].

3. Proposed Method

Fig.1 depicts the architecture of the proposed model. We set the baseline as Multispeech [22], and introduce a reference encoder [11] to extract emotion-related information. The difference between our proposed model and the baseline is the multi-style generation part that consists of several auxiliary modules.

Our proposed model takes the following processing steps. First, we generate a speaker and an emotion embedding from the speaker encoder [8] and the reference encoder [11] with a softsign function, respectively. They are concatenated and provided to a fully connected layer for combining them and reducing its dimension. The output of fully connected layer is defined as a global style embedding which is used for following processing steps. The hard alignment information defines the number of frames per each text symbol, we obtain phoneme-level F0 ($\tilde{F}_0$) by averaging the normalized reference F0 to the duration of each text symbol. One of the training objectives in the proposed model is to predict the phoneme-level F0 from the input text. We design the F0 predictor to generate appropriate normalized F0 val-
Table 1: MOS test results of the baseline and the proposed model with a 95% confidence interval.

<table>
<thead>
<tr>
<th>Emotion</th>
<th>Baseline</th>
<th>Proposed</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>3.32±0.23</td>
<td>3.47±0.23</td>
<td></td>
</tr>
<tr>
<td>Happiness</td>
<td>3.46±0.23</td>
<td>3.55±0.23</td>
<td>0.05</td>
</tr>
<tr>
<td>Sadness</td>
<td>2.39±0.28</td>
<td>2.86±0.29</td>
<td></td>
</tr>
<tr>
<td>Neutral</td>
<td>3.35±0.27</td>
<td>3.55±0.26</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>3.13±0.25</td>
<td>3.36±0.25</td>
<td></td>
</tr>
</tbody>
</table>

uses with inputs of text embedding and global style embedding. The F0 values, aligned reference in the training step or predicted one in the inference step, are encoded by a Conv1d layer to obtain F0 embedding. Subsequently, the global style embedding and the F0 embedding are concatenated to generate a multi-style embedding through a multi-style encoder. Finally, the decoder generates mel-spectrogram using the concatenation of text and multi-style embedding.

3.1. F0 generation

Since the dynamic range of the raw F0 values differs by speaker identities and emotion types, it is hard to make a model to predict the raw F0 directly. Therefore, we normalize the F0 values with the mean and standard deviation of each speaker and each emotion, and the normalized F0 values are used as the target of the F0 predictor. To extract reference F0 values from speech signals, we used the WORLD vocoder [23].

Text-dependent average F0 To train and predict F0 values from input text, the length of reference F0 values needs to be identical to that of input text. Since the target F0 values extracted from speech signals are obtained at every analysis frame, its total length is the same as mel-spectrogram frames, but different from input text length. We can compute the average F0 values within each text interval using an off-line external aligner once we define the target F0 values. However, it may bring a time alignment problem when there is an error in the F0 estimation process.

To resolve the mismatching problem caused by incorrect time alignment information, we adopt an internal aligner [20] that successfully synchronizes timing information between frame-level F0 values and the corresponding text symbols in the TTS training process. The internal aligner generates a hard alignment output, thus it represents a one-to-one correspondence between text symbols and mel-spectrogram frames. Using this information, we can obtain reference average F0 values corresponding to each phonetic symbol. In this paper, we interpret the phoneme-level F0 (F0) values as the content-dependent fine-grained style features.

F0 prediction in the inference stage Since the reference F0 values are not available in the inference stage, we need to design a model to predict them. Fig.1-(b) represents the architecture of the F0 predictor, which is inspired by the architecture of the pitch predictor in Fastspeech2 [4] and a conditional layer normalization in Adaspeech [21]. To accurately predict F0 values from input text-related embeddings, we stack 3 layers of the same architecture as shown in the figure. At this time, we utilize conditional layer normalization to provide utterance-wise global style embeddings such that the F0 predictor is able to generate F0 values based on them. After processing through the additional fully connected layer, the F0 predictor estimates appropriate normalized F0 values (F0). Because F0 is strongly related to prosody that determines speaker’s identity and emotion type, it makes sense to use F0 to generate high-quality speech samples. It is also easier for the model to predict the normalized target F0 than estimate the raw F0 values.

3.2. Multi-style encoder

Fig.1-(c) depicts the architecture of the multi-style encoder. The baseline architecture is the same as the F0 predictor, but it is not stacked and includes a softsign function into the last layer to restrict the dynamic range of output embedding. The multi-style encoder takes an input by concatenating a global style embedding and a local F0 embedding. Since the characteristics of the global style embedding and the local F0 embedding changes rapidly depending on the identity of speaker and type of emotion, the multi-style encoder needs to handle the dynamic nature of input embeddings.

Unlike just expanding global style embedding to the length of text sequence, the output of multi-style encoder varies by the type of F0 embedding. Since the F0 embedding is generated from the text-dependent average F0, we conclude that the multi-style embedding contains text-dependent local prosodic information. It is true that the global style embedding represents the entire prosodic variation of input utterance. However, it is difficult for a model to reconstruct local prosodic variation from this utterance-level global style embedding. On the contrary, our proposed model is able to reconstruct fine-grained prosodic features by combining global and local prosodic variations generated by the global style and F0 embedding.

3.3. Dynamic level F0 control

Our proposed model is able to explicitly control prosodic variations at the utterance, word and phoneme-level. As the model predicts the F0 value of the corresponding input text symbol, we can flexibly synthesize speech signals to have a desired prosodic variation or a F0 modification. For example, if we want to change the F0 of any word, we first find where the word is located in the input utterance, and then modify the corresponding F0 value as much as we want.

3.4. Training objective

To maintain stability and reliably obtain the performance, we combine several loss functions for training:

\[ \mathcal{L}_{\text{Fina}l} = \mathcal{L}_{\text{TTS}} + \mathcal{L}_{\text{IA}} + \mathcal{L}_{\text{F0}}, \]  

(1)

where \( \mathcal{L}_{\text{TTS}} \) denotes loss terms for TTS training, \( \mathcal{L}_{\text{IA}} \) for the internal aligner module, and \( \mathcal{L}_{\text{F0}} \) for the F0 predictor module.

We do not use F0 loss for 20k steps because it would be better to wait until the entire training process becomes stable. It means that we don’t use multi-style generation until 20k, but use global style embeddings. We define \( \mathcal{L}_{\text{TTS}} \) as follows:

\[ \mathcal{L}_{\text{TTS}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{rec}}^{\text{stop}} + \mathcal{L}_{\text{guide}} + \mathcal{L}_{\text{cla}^{\text{emo}}} , \]  

(2)

where \( \mathcal{L}_{\text{rec}} \) denotes L1 loss for mel-spectrogram reconstruction, \( \mathcal{L}_{\text{rec}}^{\text{stop}} \) binary cross entropy loss for stop token, \( \mathcal{L}_{\text{guide}} \) guided attention loss [2] for stable alignment in encoder-decoder attention, and \( \mathcal{L}_{\text{cla}^{\text{emo}}} \) emotion classification loss for the reference encoder.

To train the internal aligner module, we apply CTC loss from the beginning of the training process, and include KL divergence loss after 10k steps. We formulate the internal aligner...
loss as follows:

\[ L_{IA} = L_{CTC} + \lambda_{KL} L_{KL}, \]

where \( \lambda_{KL} \) is set to 0.1.

4. Experiments

4.1. Experimental setup

We trained our model with the internal Korean multi-speaker and multi-emotional dataset provided by Electronics and Telecommunications Research Institute (ETRI). There are 4 speakers (2 males and 2 females) and each speaker has 4 emotions (angry, happy, sad, neutral). Each speaker has 1 hour for each emotion so that the training data is a total of 16 hours. The validation data is a total of 2 hours, and each speaker has 7 minutes for each emotion. For data configuration, we set a sampling rate of 16 kHz, a window size of 50ms, a hop size of 12.5 ms, and a batch size of 32. We set the dimension of each embedding and emotional expressiveness. We randomly select 5 sentences from the validation dataset for each speaker and emotion so that there are 80 samples per model. Table 1 summarizes the MOS scores of the baseline and the proposed method. Our proposed model outperforms the baseline by a wide margin (3.36 vs 3.13). This difference is statistically meaningful when we calculate p-value and a 95% confidence interval. The results of sadness speech are significantly low for both baseline and proposed model. We found that there are nonverbal expressions such as sobbing in the dataset. Thus, the model generated with those nonverbal sound, which listeners regarded as poor sound quality and gave low scores. Audio samples are available online (https://kchap0118.github.io/fluenttts/).

4.2. Experimental results

To evaluate the quality of the synthesized speech signals, we performed mean opinion score (MOS) tests between the baseline and our proposed model. Seventeen listeners participated in the experiment. Listeners were asked to give a score based on the following criteria: pronunciation, natural prosodic variation and emotional expressiveness. We randomly select 5 sentences from the validation dataset for each speaker and emotion so that there are 80 samples per model. Table 1 summarizes the MOS scores of the baseline and the proposed method. Our proposed model outperforms the baseline by a wide margin (3.36 vs 3.13). This difference is statistically meaningful when we calculate p-value and a 95% confidence interval. The results of sadness speech are significantly low for both baseline and proposed model. We found that there are nonverbal expressions such as sobbing in the dataset. Thus, the model generated with those nonverbal sound, which listeners regarded as poor sound quality and gave low scores. Audio samples are available online (https://kchap0118.github.io/fluenttts/).

4.3. F0 controllability

Thanks to the F0 predictor module, our model can flexibly control the F0 values even at word-level and phoneme-level. Fig.2 illustrates the example of the word-level F0 modification for an unseen text (‘i beon yeo reum en’). We decreased (center) or increased (right) F0 values by 50 Hz to the later part (‘yeo reum en’) of the reference signal (left). We can clearly see the changes in F0 harmonics at the modified region. Fig.3 depicts the example of the phoneme-level F0 modification for an unseen text (‘u ri yu chi won’). We increased the F0 value by 50 Hz to the phoneme ‘yu’ (center) and ‘on’ (right). We can successfully modify the F0 value even in a phoneme level. We verify that our proposed model has a capability of controlling local prosodic variation. Various F0 modifications for different emotions are on our demo page.

5. Conclusion

In this paper, we proposed a text-dependent fine-grained style control method for multi-style TTS. We generated phoneme-level F0 values by an internal aligner so that we implemented a multi-style TTS model having a capability of controlling fine-grained prosody without using external modules. Since the multi-style generation part provides fine-grained prosody information to the decoder, our proposed model contains more local prosodic variation than the baseline. Moreover, our proposed model can explicitly control local prosodic variations even under rapidly changing prosodies i.e. multi-emotional TTS. Experimental results proved the performance of the proposed model. For future work, we will investigate for more effective prosody control methods even the condition under the low-resource data. Also, we consider the coverage of nonverbal sounds.

6. Acknowledgement

This research was supported by the Yonsei Signature Research Cluster Program of 2022 (2022-22-0002).
7. References


