Integrating Discrete Word-Level Style Variations into Non-Autoregressive
Acoustic Models for Speech Synthesis

Zhao-Ci Liu, Ning-Qian Wu, Ya-Jie Zhang, Zhen-Hua Ling*

National Engineering Research Center of Speech and Language Information Processing,
University of Science and Technology of China, Hefei, P.R.China
{zcliu8, wunq, zyj008}@mail.ustc.edu.cn, zhling@ustc.edu.cn

Abstract

This paper presents a method of integrating word-level style variations (WSVs) into non-autoregressive acoustic models for speech synthesis. WSVs are discrete latent representations extracted from the acoustic features of words, which have been proposed in our previous work to improve the naturalness of the Tacotron2 model. In this paper, we integrate WSVs into FastSpeech2, a non-autoregressive acoustic model. In the WSV extractor, a Gumbel-Sigmoid activation function is introduced for WSV representation and is compared with the original Gumbel-Softmax activation by experiments. The WSV predictor utilizes the word embeddings provided by BERT and has a non-autoregressive structure to be compatible with FastSpeech2. Experimental results show that our proposed method with the Gumbel-Sigmoid activation achieved better objective performance on F0 prediction than the FastSpeech2 baseline and the method using the Gumbel-Softmax activation. The subjective performance of our proposed models was also significantly better than the FastSpeech2 baseline.

Index Terms: speech synthesis, FastSpeech2, Gumbel function, representation learning

1. Introduction

The naturalness of speech synthesis has improved greatly with the rapid evolution of neural network-based acoustic models [1,2] and neural vocoders [3,4] in recent years. However, there is still a clear gap between the naturalness of synthetic speech and that of human recordings when a corpus with rich expressiveness is used. While, some application scenarios, such as audiobooks, voice assistants and dialogue interactions, require that synthetic speech should contain rich expressiveness and stylistic variations. Moreover, one utterance may correspond to different prosody characters depending on the speaker’s intention or contextual information, which also increases the difficulty of synthesizing expressive speech.

Intuitively, semantic information affects prosody characters, such as stress and intonation. Pre-trained language models, e.g., Bidirectional Encoder Representations from Transformers (BERT) [5], can derive semantic-related word (or subword) representations and several recent studies have applied the contextual representations extracted by BERT to the sequence-to-sequence acoustic modeling of speech synthesis [6–8] and improved the naturalness of synthetic speech.

Another approach to expressive speech synthesis is to learn latent style representations from speech signals. The Global Style Token (GST) model [9] decomposed the sentence-level style embedding vector into a fixed number of style tokens and the text-predicted global style token (TP-GST) model [10] further predicted the style tokens from text sequences and achieved better subjective quality of synthetic speech than the Tacotron [1] baseline on audiobook datasets. Considering that prosody by nature has a hierarchical structure, involving sentences, phrases, words, and phonemes, the latent representations at a finer granularity than sentences have also been studied [11–14].

In our previous work [14], a latent representation, named word-level style variation (WSVs), was extracted and predicted to improve the naturalness of the Tacotron2 baseline. However, all the models mentioned above adopted autoregressive sequence-to-sequence architectures which constrained the inference efficiency. On the other hand, some non-autoregressive speech synthesis models [15–17] have been proposed and have greatly improved the speed of speech synthesis. Recently, there are several studies on combining latent representations with non-autoregressive sequence-to-sequence architectures. Du et al. [18] modeled prosody using a Gaussian mixture model, but at the phoneme level. Zhang et al. [19] focused on extracting prosodic representations using information bottlenecks, and the baseline synthesis model differed from the complete configuration of FastSpeech2.

Therefore, this paper proposes to introduce the WSVs proposed in our previous work [14] into FastSpeech2 and designs the corresponding non-autoregressive WSV predictor. The WSV extractor produces discrete WSVs from the Mel-spectrogram of each word. The WSV predictor is a Transformer-based neural network and predicts WSVs utilizing both phoneme sequences and the word embeddings extracted from a pre-trained BERT [5] model. In our previous study [14], WSVs were categorical representations obtained by Gumbel-Softmax [20]. In this paper, we also propose to obtain WSVs of Bernoulli representations using a Gumbel-Sigmoid [20] activation function. Experimental results show that our proposed method based on Gumbel-Sigmoid outperformed the FastSpeech2 baseline and the method using Gumbel-Softmax.

2. Baseline FastSpeech2 Model

FastSpeech2 [16] is an improved version of the FastSpeech model [15]. It adopts a non-autoregressive architecture and synthesize speech significantly faster and with a comparable quality compared to previous autoregressive models, e.g., Tacotron2 [2]. It solves the one-to-many mapping problem in speech synthesis by introducing prosody features of speech (i.e., pitch, energy and duration) as conditional inputs of its decoder. These prosody features are extracted from natural speech in the training phase and are predicted from text in the inference phase. Compared with original configurations in the FastSpeech2 paper, the pitch predictor and energy predictor in our implementation have 4 convolutional layers and the kernel

* Corresponding author. This work was supported in part by the National Nature Science Foundation of China under Grant 61871358.
FastSpeech-based Synthesizer  
WSV Predictor
WSV ... A WSV predictor is necessary to predict WSVs from input text. The WSV predictor in our previous work [14] has an

3. Proposed Method

3.1. Model architecture

As shown in Fig. 1, our proposed model can be divided into three main parts: a WSV extractor, a WSV predictor, and a FastSpeech2-based synthesizer. The synthesizer produces the Mel-spectrum of the target sentence with the WSVs given by the WSV extractor or the WSV predictor. The word-level style embeddings are derived by calculating the weighted sum of word-level style tokens using WSVs. These style embeddings are repeated to phone-level and are then added to the outputs of the text encoder in the synthesizer for further processing using the variance adapter and the decoder.

3.1.1. WSV extractor

The structure of the WSV extractor is the same as the one in our previous work [14], which consists of a reference encoder and an attention module. The Mel-spectrogram of each word is passed through the reference encoder to obtain a word-level reference embedding. Here, the word boundaries are determined using the alignment between words and phonemes together with the phoneme boundaries given by forced alignment. Then, the single-headed attention module encodes the word-level reference embedding into a WSV vector. The word-level reference embeddings act as queries in the attention mechanism, while the randomly initialized word-level style tokens act as both keys and values. These word-level style tokens are trained to learn the word-level rhythmic or stylistic features of speech. The token weights of a word form its word-level style variation (WSV) vector, and the style tokens are weighted by the WSV to obtain the word-level style embedding vector.

In our previous work [14], a Gumbel-Softmax activation function is adopted by the attention module, which makes the calculated WSVs close to one-hot vectors. Thus, predicting WSVs can be considered as a classification task rather than a regression task. In this paper, a Gumbel-Sigmoid activation function is also employed for comparison. A Sigmoid function is adopted by the attention module, which makes the calculated WSVs close to one-hot vectors. Thus, predicting WSVs describe local acoustic properties. At the inference stage, selecting a reference speech to derive these local descriptions is infeasible. Thus, A WSV predictor is necessary to predict WSVs from input text.

The WSV predictor in our previous work [14] has an
autoregressive structure, which constrains the inference efficiency of the FastSpeech2-based synthesizer. Here, a non-autoregressive WSV predictor is designed as shown in Fig. 2. It accepts two types of inputs. One is the phoneme sequence, which is directly processed using the text encoder in the synthesizer. The other is the original text sequence, which is converted to a token sequence and further encoded by a pre-trained BERT model. Given token-to-word and phoneme-to-word alignment relationships, both outputs of the BERT model and the text encoder are converted to word-level representations by average pooling within each word. These two representations are then concatenated and sent to the non-autoregressive decoding module.

The decoding module contains a single-layer Feed Forward Transformer Block (FFTBlock) [15], and two linear layers before and after to change the channel size. The output layer activation function is either Sigmoid or Softmax according to the activation function used in the WSV predictor. Compared with the continuous style embedding vectors, WSVs had fewer dimensions and are binarized, which are more appropriate for prediction according to the results of our previous study [14].

3.2. Model training and inference

The model is trained by two stages. At the first stage, the switch in Fig. 1 turns to right. So that the WSV extractor and the FastSpeech2-based synthesizer are trained simultaneously. For each utterance in the training set, its Mel-spectrogram and phoneme sequence are sent to the WSV extractor and the synthesizer, respectively. The training criterion for this part is the same as FastSpeech2, including the loss of predicting Mel-spectrogram and the losses of phoneme duration, energy, and pitch predictors. At the second stage, the switch turns to left and the WSV predictor is trained. The true WSVs are obtained by binarizing the output of the WSV extractor, and Gumbel sampling is applied during this process. The loss function of the WSV predictor is defined as the cross-entropy loss between the true WSVs and the predicted WSVs. When training the WSV predictor, all other parameters of the model are fixed.

At the inference stage, the switch turns to left. So that the WSV predictor and the synthesizer are connected to form a complete acoustic model, which predicts the Mel-spectrogram from the input text and phoneme sequence of the sentence to be synthesized. The predicted Mel-spectrogram is further passed through a Parallel WaveGAN (PWG) vocoder [3] to reconstruct speech waveforms.

4. Experiments

4.1. Experimental conditions

The Blizzard Challenge 2019 dataset [21], which contains approximately 8-hour talkshow speech from a Chinese male speaker, was used for our experiments. 492 sentences were randomly taken out from the dataset (245 sentences for validation, and 247 sentences for test), and the remaining ones formed the training set. The acoustic features used in the experiments were 80-dimensional Mel-spectrogram, with a frame shift of 12.5 ms and a frame length of 50 ms. Phoneme sequences (i.e., initials and finals in Chinese) were used as the input of the text encoder in FastSpeech2. Each phoneme was represented as the concatenation of a phoneme identity embedding vector, a tone embedding vector and a prosodic position embedding vector [14]. The phoneme durations of training data came from an internal forced alignment tool based on HMM. F0 values were extracted [22] and interpolated using an exponential function at unvoiced segments. The Parallel WaveGAN vocoder was trained on the same dataset.

In addition to the FastSpeech2 baseline, a Tacotron2 baseline model and our previously proposed WSV modeling method based on Tacotron2 [14] were also built for comparison. Two proposed models, FS2WSV_Ber and FS2WSV_Cat, were built by adopting Gumbel-Sigmoid and Gumbel-Softmax in the WSV extractor, respectively.

The reference encoder in the WSV extractor had the same architecture and hyperparameters as the ones of GST-Tacotron [9], except that the input Mel-spectrogram was cropped. Only the lowest 10 dimensions of the 80-dimensional Mel-spectrogram were kept, which covered roughly 0-370Hz, in order to make the extracted WSVs decoupled from the text content and focus on prosody. The dimension of WSVs was set as $K = 10$. The temperature parameter in Gumbel activation was set to 1. The configurations of the FastSpeech2 synthesizer were the same as its original paper except the modifications mentioned in Section 2. At the first stage of model training, the WSV extractor and the synthesizer were updated up to 200,000 steps using the same strategy of learning rate warmup and decay as FastSpeech2.

In the WSV predictor, a pretrained Chinese BERT BASE model released by Huggingface-Transformer [23] was adopted. The BERT model consisted of 12 Transformer layers, with 768 hidden units and 12 self-attention heads in each layer. The output activations of the 11-th layer were used as token embeddings. The FFTBlock in the WSV predictor had a hidden size of 128. The WSV predictor was trained for 4,000 steps with Adam optimizer [24] and the learning rate exponentially decayed from $10^{-4}$ to $10^{-5}$. Other model parameters, including BERT, were fixed when training the WSV predictor. These models were implemented in PyTorch and were trained on a GeForce GTX 2080Ti GPU with a batchsize of 32.

4.2. Subjective evaluation

Three preference tests on naturalness were conducted to compare our proposed models and the FastSpeech2 baseline. 20 sentences were randomly chosen from the test set and were synthesized by different models. At least 10 native Chinese

---

<table>
<thead>
<tr>
<th>Model</th>
<th>MOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground truth</td>
<td>4.68 ± 0.08</td>
</tr>
<tr>
<td>FastSpeech2</td>
<td>3.98 ± 0.11</td>
</tr>
<tr>
<td>FS2WSV_Ber</td>
<td>4.20 ± 0.08</td>
</tr>
</tbody>
</table>

Table 1: Average preference scores on naturalness among different models, where N/P means “no preference” and $p$ denotes the p-value of a t-test between two models.

Table 2: The naturalness mean opinion scores (MOS) of different models with 95% confidence intervals.
Table 3: The objective evaluation results of natural speech and the speech generated by different models on the test set. Here, “Corr” means correlation, “SD” means standard deviation. The models labeled with * denote that the WSVs were extracted from the recordings of test utterances instead of being predicted using the WSV predictor.

<table>
<thead>
<tr>
<th>System</th>
<th>F0 Corr</th>
<th>F0 RMSE (Hz)</th>
<th>F0 SD (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>-</td>
<td>-</td>
<td>51.8</td>
</tr>
<tr>
<td>FastSpeech2</td>
<td>0.556</td>
<td>42.3</td>
<td>40.2</td>
</tr>
<tr>
<td>FS2WSV_Ber*</td>
<td>0.753</td>
<td>32.5</td>
<td>41.2</td>
</tr>
<tr>
<td>FS2WSV_Cat*</td>
<td>0.718</td>
<td>35.3</td>
<td>40.5</td>
</tr>
<tr>
<td>FS2WSV_Ber</td>
<td>0.582</td>
<td>41.0</td>
<td>40.3</td>
</tr>
<tr>
<td>FS2WSV_Cat</td>
<td>0.559</td>
<td>42.8</td>
<td>40.9</td>
</tr>
</tbody>
</table>

Table 4: The average accuracies and macro F1 values of WSV predictors in different models on the test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
<th>Macro F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS2WSV_Ber</td>
<td>76.1</td>
<td>69.7</td>
</tr>
<tr>
<td>(FFTBlock → LSTM)</td>
<td>73.2</td>
<td>66.9</td>
</tr>
<tr>
<td>FS2WSV_Cat</td>
<td>26.6</td>
<td>25.9</td>
</tr>
<tr>
<td>(FFTBlock → LSTM)</td>
<td>26.1</td>
<td>25.9</td>
</tr>
</tbody>
</table>

listeners took part in each test and the results are shown in Table 1. We can see that both proposed models achieved significantly higher preference scores than the FastSpeech2 baseline. Between these two proposed models, using the Gumbel-Sigmoid activation function was slightly better than using Gumbel-Softmax, but the superiority was insignificant.

A mean opinion score (MOS) test was also conducted to measure the naturalness of FS2WSV_Ber model and the FastSpeech2 baseline. The MOS scale was from 1 to 5 (with intervals of 0.5), when 1 indicated completely unnatural and 5 indicated completely natural. 10 native Chinese listeners took part in the test. The speech recovered from the ground-truth Mel-spectrograms with the same Parallel WaveGAN vocoder was also evaluated for reference. The results are shown in Table 2. Be consistent with the results in Table 1, our proposed FS2WSV_Ber model achieved significantly higher naturalness than the FastSpeech2 baseline.

4.3. Analysis experiments

(1) Accuracy of F0 prediction. F0 correlation, F0 root mean square error (RMSE) and F0 standard deviation (SD) were used as the metrics for objective evaluation. To calculate F0 correlation and F0 RMSE, the forced alignment algorithm was first applied to align the synthesized speech with the phoneme sequences. Then, F0 values were extracted and interpolated at unvoiced segments. Afterward, the average F0 of each phoneme was calculated to further derive the F0 correlation [25] and the F0 RMSE with that of ground truth speech. The F0 SDs were calculated separately at frame level using the F0 values of all voiced frames. Table 3 shows that the FS2WSV_Ber* and FS2WSV_Cat* models achieved significantly higher accuracy of F0 prediction than other models, which demonstrates the effectiveness of the extracted WSVs on describing local prosody variations. It also can be seen that the 10 Bernoulli variables have better representation ability than the one-hot-category variable. The F0 correlation and F0 RMSE of our proposed models, especially FS2WSV_Ber was better than the results of FastSpeech2 baseline. The F0 SDs of all models were similar and much smaller than the natural one, which may be due to the sentence-level F0 normalization in the variance adaptor of FastSpeech2 [16].

(2) Accuracy of WSV prediction. We evaluated the accuracies and macro F1 values of WSV prediction in our proposed models and compared their performance with using the original LSTM-based WSV predictor [14]. The results are shown in Table 4. Since FS2WSV_Ber and FS2WSV_Cat adopted different types of WSV representations, their results were not comparable. From this table, we can see that using FFTBlock instead of LSTM in our proposed models didn’t harm the performance of WSV prediction, but can better support parallel computing at the inference stage.

(3) Comparison with Tacotron2 models. A preference test on naturalness was conducted to compare FS2WSV_Cat with Tacotron2_WSVCat, which integrated categorical WSVs into Tacotron2 in our previous work [14]. The test configurations were the same as the ones in Section 4.2. We can see that FS2WSV_Cat achieved an insignificantly higher preference score than Tacotron2_WSVCat. We also measured the real time factors (RTFs) of Mel-spectrogram generation using different models and the results are shown in Table 5. We can see that our proposed models ran slightly slower than the FastSpeech2 baseline due to the additional WSV predictor, but it was still more than 12 times faster than Tacotron2_WSV_Cat due to the non-autoregressive and non-recursive architecture.

5. Conclusion

This paper has proposed a method of integrating fine-grained latent representations into the non-autoregressive FastSpeech2 model for speech synthesis. The representations are word-level style variations (WSVs) extracted from the acoustic features within words using attention with either Gumbel-Softmax or Gumbel-Sigmoid activation. A pre-trained BERT model is introduced as an additional text input of the WSV predictor to provide semantic-related information. The proposed model is fully non-autoregressive at synthesis time. Experimental results show that our proposed method achieved better objective and subjective performance than the baseline FastSpeech2 model. To improve the WSV predictor with better model architecture and wider context input will be the task of our future work.
6. References


