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

Pre-trained Text Embeddings for Enhanced Text-to-Speech Synthesis

Tomoki Hayashi[1], *Shinji Watanabe*[2], *Tomoki Toda*[1], *Kazuya Takeda*[1]
Shubham Toshniwal[3], *Karen Livescu*[3]
[1]Nagoya University, Japan
[2]Johns Hopkins University, USA
[3]Toyota Technological Institute at Chicago, USA
hayashi.tomoki@g.sp.m.is.nagoya-u.ac.jp, shinjiw@jhu.edu,
tomoki@icts.nagoya-u.ac.jp, kazuya.takeda@nagoya-u.ac.jp
{shtoshni, klivescu}@ttic.edu

Abstract

We propose an end-to-end text-to-speech (TTS) synthesis model that explicitly uses information from pre-trained embeddings of the text. Recent work in natural language processing has developed self-supervised representations of text that have proven very effective as pre-training for language understanding tasks. We propose using one such pre-trained representation (BERT) to encode input phrases, as an additional input to a Tacotron2-based sequence-to-sequence TTS model. We hypothesize that the text embeddings contain information about the semantics of the phrase and the importance of each word, which should help TTS systems produce more natural prosody and pronunciation. We conduct subjective listening tests of our proposed models using the 24-hour LJSpeech corpus, finding that they improve mean opinion scores modestly but significantly over a baseline TTS model without pre-trained text embedding input.

Index Terms: speech synthesis, end-to-end, pre-trained text embedding

1. Introduction

Text-to-speech (TTS) synthesis is the problem of generating speech given input text. TTS is a key component in various applications such as providing navigation directions in smartphones and cars, interactive interfaces such as smart assistants, and in systems designed to assist individuals with disabilities. Typical statistical TTS systems [1–3] consist of many modules such as a natural language parser, duration model, acoustic model, and vocoder, utilizing many handcrafted language features developed by experts. This factorization of TTS systems allows training each module separately, but results in errors propagating from one component to subsequent components.

As a result of the development of effective deep learning techniques, end-to-end TTS (E2E-TTS) systems have become viable and have even started replacing conventional TTS systems in production [4–8]. In general, an E2E-TTS system has two modules: a feature generation module and a waveform synthesis module. The feature generation module is typically modeled as a sequence-to-sequence (seq2seq) neural network [9–11], which converts a given text into a sequence of acoustic features such as spectrograms or mel cepstra. Since this network directly converts a sequence of characters or phones into acoustic features, it does not require complex text analysis or duration modeling of each phone, as in conventional TTS systems. The waveform synthesis module is also a neural network [12–16] which converts a given sequence of acoustic features into a waveform signal, avoiding a signal processing based vocoder [17, 18] that may degrade the quality of synthesized speech due to simplifying assumptions about human speech generation.

To further improve the pronunciation or controllability of speech produced by E2E-TTS models, a number of extensions have been studied. In [19], a speaker embedding is utilized to allow the model to generate speech of various speakers. Jia *et al.* [20] utilized the speaker embedding learnt in a speaker verification model. Along similar lines, [21, 22] proposed the use of a style embedding to make generated speech more expressive. These approaches have improved the quality of generated speech, but they require the collection of specific types of speech samples to learn the embeddings. On the other hand, Yasuda *et al.* [23] used additional accent information to extract an accent embedding, which forms an additional input to the encoder. This approach greatly improves the quality of generated Japanese speech, but it requires accent information that is difficult to obtain automatically.

In this study, we propose a novel E2E-TTS model that explicitly utilizes text-context information. Our approach is inspired by the recent greatly improved performance of sentence representation models in natural language processing [24, 25]. In particular we propose an approach that relies on Bidirectional Encoder Representations from Transformers (BERT) [24]. Our proposed approach uses text-context embeddings computed by a pre-trained BERT model as an additional input for a Tacotron2-based seq2seq synthesis model [7]. We hypothesize that these contextual embeddings contain useful information such as the positiveness of the sentence and the importance of each word in the sentence. Therefore, it is expected that the use of the additional context information will improve the pronunciation and expressiveness of the generated speech. Furthermore, a BERT embedding can be extracted from a given text without any additional inputs, and BERT itself is trained using just unlabelled text, thus not requiring to collect various types of speech samples.

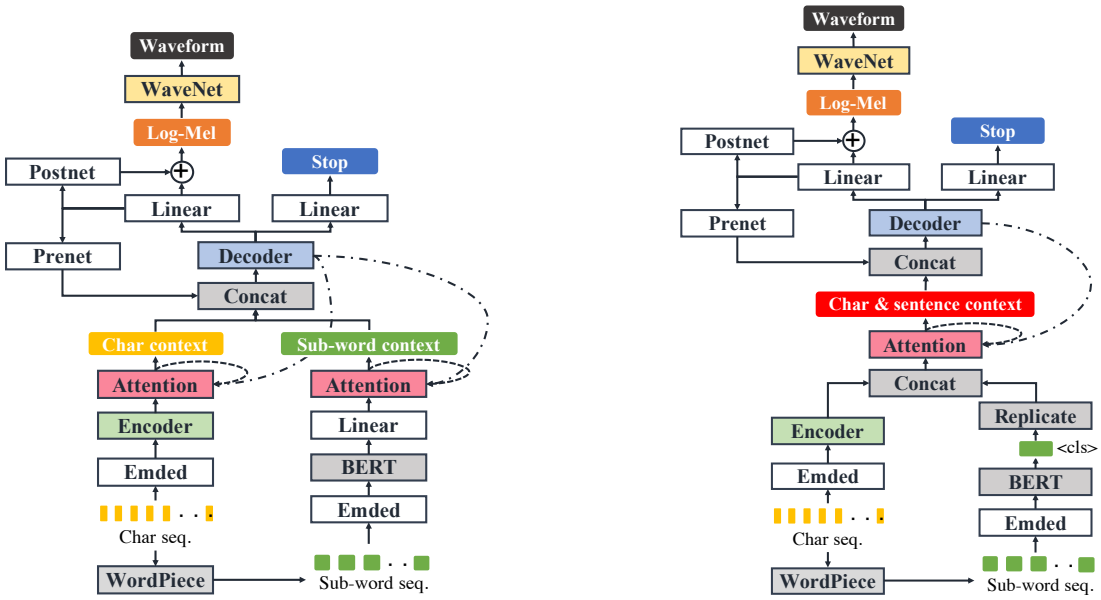
We conduct a subjective evaluation using the LJSpeech database [26], which consists of 24 hours of English speech from a single speaker. Our experimental results show that the proposed model improves the naturalness of the generated speech.

2. Proposed Method

2.1. System overview

An overview of the proposed method is shown in Fig. 1. In this study, we propose two models—a subword-level model and a phrase-level model—based on the granularity of the text-context features being used. Both variants consist of three neural networks: a text-context extraction network, a spectrogram prediction network, and a vocoder network. The text-context extraction network is a pre-trained model that extracts contex-

http://dx.doi.org/10.21437/Interspeech.2019-3177



(a) *Subword-level model* (b) *Phrase-level model*

Figure 1: *Overview of the proposed methods.*

tual embeddings from a given input text. The spectrogram prediction network is a seq2seq model which generates log mel-spectrogram features given the inputs: the sequence of characters and the output of the text-context extraction network. In the case of the subword-level model, a sequence of subword embeddings is used as an additional input for the spectrogram prediction network. In the case of the phrase-level model, the extracted phrase embedding is used similarly to a speaker embedding [20]. The vocoder network is a deep convolutional neural network (CNN) which generates a waveform from given acoustic features. We describe each module in detail in the following sections.

2.2. Text-context extraction network

We use a pre-trained BERT model [24], a deep bi-directional Transformer [27] that computes a deep representation by considering both past and future contexts, as the text-context extraction network.

The length-L sequence of characters $\mathbf{c} = \{c_1, c_2, \ldots, c_L\}$ is decomposed converted into a $(L' + 1)$-length sequence of "wordpieces" [28] that consists of a special symbol and a series of subwords $\mathbf{c}' = \{\text{<cls>}, c'_1, c'_2, \ldots, c'_{L'}\}$. Here, <cls> represents the special symbol for a sentence classification task which is added to the beginning of the sequence. The input $\mathbf{X} \in \mathbb{R}^{(L'+1)\times D}$ to BERT is the sum of D-dimensional token, segmentation, and positional embeddings. Given the input, the B-block Transformer operates as follows:

$$\mathbf{R}^{(b)} = \text{TransformerBlock}^{(b)}\left(\mathbf{R}^{(b-1)}\right), \tag{1}$$

where $\mathbf{R}^{(b)} = \{\mathbf{r}^{(b)}_{\text{<cls>}}, \mathbf{r}^{(b)}_1, \mathbf{r}^{(b)}_2, \ldots, \mathbf{r}^{(b)}_{L'}\} \in \mathbb{R}^{(L'+1)\times D}$ represents the outputs of the b-th block, $b \in \{1, 2, \ldots, B\}$ represents the index of the blocks, and $\mathbf{R}^{(0)}$ corresponds to input $\mathbf{X}$. The $\text{TransformerBlock}(\mathbf{X})$ is calculated as follows:

$$\mathbf{o} = \text{MultiHeadAttention}(\mathbf{X}), \tag{2}$$
$$\mathbf{g} = \text{LayerNorm}(\mathbf{o} + \mathbf{x}), \tag{3}$$
$$\mathbf{r} = \text{LayerNorm}\left(\text{FNN}(\mathbf{g}) + \mathbf{g}\right), \tag{4}$$

where $\text{LayerNorm}(\cdot)$ represents a layer normalization operation [29], $\text{FNN}(\cdot)$ represents a linear layer with a Gaussian error linear unit (GELU) activation function [30] and learnable weight/bias parameters, and $\text{MultiHeadAttention}(\mathbf{X})$ represents H-head attention [27], which is calculated as follows:

$$\mathbf{Q}_i = \mathbf{X}\mathbf{W}^{(Q)}_i,\ \mathbf{K}_i = \mathbf{X}\mathbf{W}^{(K)}_i,\ \mathbf{V}_i = \mathbf{X}\mathbf{W}^{(V)}_i, \tag{5}$$
$$\mathbf{o}'_i = \text{Softmax}\left(\mathbf{Q}_i\mathbf{K}_i^\top/\sqrt{D_k}\right)\mathbf{V}_i, \tag{6}$$
$$\mathbf{o} = \text{Concat}(\mathbf{o}'_1, \mathbf{o}'_2, \ldots, \mathbf{o}'_H)\mathbf{W}^{(O)}, \tag{7}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V} \in \mathbb{R}^{(L'+1)\times D_k}$ represent query, key, and value transformation of the inputs, respectively. $\mathbf{W}^{(Q)}_i$, $\mathbf{W}^{(K)}_i$, $\mathbf{W}^{(V)}_i \in \mathbb{R}^{D\times D_k}$, and $\mathbf{W}^{(O)} \in \mathbb{R}^{HD_k\times D}$ represent learnable matrix parameters, $\text{Concat}(\cdot)$ represents a vector concatenation operation, $i \in \{1, 2, \ldots, H\}$ represents the index of the head, and D_k represents the dimension of hidden states.

Finally, we use the outputs of the final (B-th) Transformer blocks $\mathbf{R}^{(B)} = \{\mathbf{r}^{(B)}_{\text{<cls>}}, \mathbf{r}^{(B)}_1, \mathbf{r}^{(B)}_2, \ldots, \mathbf{r}^{(B)}_{L'}\} \in \mathbb{R}^{(L'+1)\times D}$ as an additional input to the spectrogram prediction network. For the subword-level model, we use $\{\mathbf{r}^{(B)}_1, \mathbf{r}^{(B)}_2, \ldots, \mathbf{r}^{(B)}_{L'}\}$ as additional inputs, whereas for the phrase-level model we use just $\mathbf{r}^{(B)}_{\text{<cls>}}$ as an additional input.

2.3. Spectrogram prediction network

A Tacotron2 [7] based seq2seq network, which consists of encoder, attention, and decoder networks, is used as the spectrogram prediction network. As an extension of the original Tacotron2, we also use text-context embeddings extracted by the text-context extraction network. We first present the subword-level model and then the modifications we make to obtain the phrase-level model.

The log mel-spectrogram $\mathbf{y}_t$ and the probability of the end of sequence $p^{(\text{eos})}_t$ are calculated using a given sequence of characters $\mathbf{c} = \{c_1, c_2, \ldots, c_L\}$ and a sequence of subword embeddings $\{\mathbf{r}^{(B)}_1, \mathbf{r}^{(B)}_2, \ldots, \mathbf{r}^{(B)}_{L'}\}$ as follows:

$$\mathbf{h}^{(\text{char})}_l = \text{Encoder}_l(\mathbf{c}), \tag{8}$$

$$\mathbf{h}_{l'}^{(\mathrm{sub})} = \mathrm{LinB}\left(\mathbf{r}_{l'}^{(B)}\right), \tag{9}$$

$$\mathbf{z}_t^{(\mathrm{char})} = \mathrm{Attention}^{(\mathrm{char})}\left(\mathbf{q}_{t-1}, \mathbf{H}^{(\mathrm{char})}, \mathbf{a}_{t-1}^{(\mathrm{char})}\right), \tag{10}$$

$$\mathbf{z}_t^{(\mathrm{sub})} = \mathrm{Attention}^{(\mathrm{sub})}\left(\mathbf{q}_{t-1}, \mathbf{H}^{(\mathrm{sub})}, \mathbf{a}_{t-1}^{(\mathrm{sub})}\right), \tag{11}$$

$$\mathbf{u}_{t-1} = \mathrm{Prenet}\left(\mathbf{y}_{t-1}\right), \tag{12}$$

$$\mathbf{z}_t = \mathrm{Concat}\left(\mathbf{z}_t^{(\mathrm{char})}, \mathbf{z}_t^{(\mathrm{sub})}, \mathbf{u}_{t-1}\right), \tag{13}$$

$$\mathbf{q}_t = \mathrm{Decoder}\left(\mathbf{z}_t, \mathbf{q}_{t-1}\right), \tag{14}$$

where $\mathrm{Encoder}(\cdot)$, $\mathrm{Attention}(\cdot)$, and $\mathrm{Decoder}(\cdot)$ represent an encoder, attention, and decoder network, respectively, $\mathrm{LinB}(\cdot)$ represents a linear layer with learnable weight and bias parameters, $\mathbf{H}$ represents a sequence $\{\mathbf{h}_1, \mathbf{h}_2, \dots\}$, $\mathbf{a}_t^{(\mathrm{char})} = \{a_{t,1}, a_{t,2}, \dots, a_{t,L}\}$ and $\mathbf{a}_t^{(\mathrm{sub})} = \{a_{t,1}, a_{t,2}, \dots, a_{t,L'}\}$ represent attention weights for the sequence of character embeddings and that of subword embeddings, respectively, $\mathrm{Prenet}(\cdot)$ is a shallow feed-forward network to convert the outputs before feedback to the decoder. Then, the final outputs are calculated as follows:

$$p_t^{(\mathrm{eos})} = \mathrm{Sigmoid}(\mathrm{LinB}(\mathbf{q}_t)), \tag{15}$$

$$\hat{\mathbf{y}}_t^{(\mathrm{before})} = \mathrm{LinB}(\mathbf{q}_t), \tag{16}$$

$$\hat{\mathbf{Y}}^{(\mathrm{after})} = \hat{\mathbf{Y}}^{(\mathrm{before})} + \mathrm{Postnet}\left(\hat{\mathbf{Y}}^{(\mathrm{before})}\right), \tag{17}$$

where $\mathrm{Postnet}(\cdot)$ is a CNN to refine the network outputs, $\hat{\mathbf{Y}}$ represents a sequence $\{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, \dots, \hat{\mathbf{y}}_T\}$, and $\hat{\mathbf{Y}}^{(\mathrm{before})}$ and $\hat{\mathbf{Y}}^{(\mathrm{after})}$ represent the predicted mel-spectrogram before and after refinement by Postnet, respectively. For the subword-level model, we use forward attention with a transition agent [31] as $\mathrm{Attention}(\cdot)$ in Eqs. (10,11) in order to learn a reasonable diagonal attention pattern.

In the case of the phrase-level model, the phrase embedding $\mathbf{r}_{<\mathrm{cls}>}^{(B)}$ is used instead of subword embeddings. Eqs. (9, 11) are not used, and Eqs. (8, 13) are replaced with the following equations, respectively:

$$\mathbf{h}_l^{(\mathrm{char})} = \mathrm{Concat}\left(\mathrm{Encoder}_l(\mathbf{c}), \mathbf{r}_{<\mathrm{cls}>}^{(B)}\right), \tag{18}$$

$$\mathbf{z}_t = \mathrm{Concat}\left(\mathbf{z}_t^{(\mathrm{char})}, \mathbf{u}_{t-1}\right). \tag{19}$$

We use location-sensitive attention [7] as $\mathrm{Attention}(\cdot)$ in Eq. (10) for the phrase-level model.

The whole network is trained to minimize the following loss function:

$$\begin{aligned}\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\Big\{ & \mathcal{L}_{\mathrm{mse}}\left(\hat{\mathbf{y}}_t^{(\mathrm{before})}, \mathbf{y}_t\right) + \mathcal{L}_{\mathrm{mse}}\left(\hat{\mathbf{y}}_t^{(\mathrm{after})}, \mathbf{y}_t\right) \\ & + \mathcal{L}_{\mathrm{eos}}(p_t^{(\mathrm{eos})}, s_t) + \mathcal{L}_{\mathrm{l1}}\left(\hat{\mathbf{y}}_t^{(\mathrm{before})}, \mathbf{y}_t\right) + \mathcal{L}_{\mathrm{l1}}\left(\hat{\mathbf{y}}_t^{(\mathrm{after})}, \mathbf{y}_t\right) \\ & + \lambda\left(\mathcal{L}_{\mathrm{att}}\left(\mathbf{a}_t^{(\mathrm{char})}\right) + \mathcal{L}_{\mathrm{att}}\left(\mathbf{a}_t^{(\mathrm{sub})}\right)\right)\Big\},\end{aligned} \tag{20}$$

where $\mathcal{L}_{\mathrm{mse}}$ represents a mean square error for the predicted log mel-spectrogram, $\mathcal{L}_{\mathrm{l1}}$ represents an L1 loss for the predicted log mel-spectrogram, $\mathcal{L}_{\mathrm{eos}}$ represent binary cross entropy for the probability of the end of the sequence, $\mathcal{L}_{\mathrm{att}}$ represents guided attention loss [32], and λ is a hyperparameter for weighing the attention-based loss terms.

2.4. Vocoder network

We use a variant of WaveNet vocoder [12, 33] which uses acoustic features (e.g., log mel-spectrogram) as auxiliary features. The conditional probability of a length-N waveform signal $\mathbf{x} = \{x_{[1]}, x_{[2]}, \dots, x_{[N]}\}$ given the acoustic features $\mathbf{y}$ is factorized as follows:

$$p(\mathbf{x}|\mathbf{y}) = \prod_{n=1}^{N} p(x_{[n]}|x_{[1:n-1]}, \mathbf{y}), \tag{21}$$

where $n \in \{1, 2, \dots, N\}$ represents the sample index in the waveform and $x_{[1:n-1]}$ represents the sub-sequence $\{x_{[1]}, x_{[2]}, \dots, x_{[n-1]}\}$. WaveNet approximates the conditional probability above by ignoring the effect of past samples beyond a receptive field of size F as follows:

$$\mathrm{WaveNet}(\mathbf{x}, \mathbf{y}) \simeq p(x_{[n]}|x_{[n-F:n-1]}, \mathbf{y}). \tag{22}$$

WaveNet consists of many residual blocks, each of which consists of 2×1 dilated causal convolutions, a gated activation function and 1×1 convolutions. The gated activation function is calculated as follows:

$$\begin{aligned}\mathbf{z}^{(\mathrm{gated})} = & \mathrm{Tanh}\left(\mathbf{W}_{f,k}^{(\mathrm{dil})} * \mathbf{x} + \mathbf{W}_{f,k}^{(1\times1)} * f(\mathbf{y})\right) \odot \\ & \mathrm{Sigmoid}\left(\mathbf{W}_{g,k}^{(\mathrm{dil})} * \mathbf{x} + \mathbf{W}_{g,k}^{(1\times1)} * f(\mathbf{y})\right),\end{aligned} \tag{23}$$

where $\mathbf{W}$ represents trainable convolution filters, $\mathbf{W}^{(\mathrm{dil})} * \mathbf{x}$ represents a dilated causal convolution, $\mathbf{W}^{(1\times1)} * f(\mathbf{y})$ represents a 1×1 convolution, $\odot$ represents element-wise multiplication, subscript k is the layer index, subscripts f and g represent the filter and gate, respectively, and $f(\cdot)$ represents the function that transforms features $\mathbf{y}$ to have the same time resolution as the input waveform. The waveform signal is quantized into 8 bits using the μ-law algorithm [34] and converted into a sequence of 256 dimensional (= 8 bits) one-hot vectors. In order to reduce the noise caused by prediction errors, a time-invariant noise shaping technique [35] is applied.

3. Experimental Evaluation

3.1. Training Setup

Table 1 describes all the model and training details. The spectrogram prediction network was constructed using the open-source speech processing toolkit ESPnet [36] with a PyTorch [37] backend. The WaveNet vocoder was trained with the natural speech from training data using the PyTorch implementation.[1] Since the additional text context affects only the spectrogram prediction component, the WaveNet vocoder module is a constant across all the models. We used the pre-trained parameters of the BERT-large uncased model.[2]

For the experiments, we used the LJSpeech database [26], which consists of 24 hours of English speech from a single speaker. We used 12,600 utterances for training, 250 utterances for validation, and 250 utterances for evaluation. We trained three models: the baseline model without any text context (Baseline), the phrase-level model (Phrase), and the subword-level model (Subword). We decided the best parameters by monitoring the loss for the validation set and used the first 80 utterances of the evaluation set for a subjective evaluation. Generated samples are available publicly.[3]

3.2. Evaluation

We conduct a subjective evaluation using eleven subjects consisting of five Japanese speakers and six English speakers. We use the following two evaluation metrics:

[1] https://github.com/kan-bayashi/PytorchWaveNetVocoder.

[2] https://github.com/google-research/bert.

[3] https://kan-bayashi.github.io/Taco2withBERT.

Table 1: *Model and training configuration*

Sampling rate	22,050 Hz
Window size	46.4 ms (1,024 pt)
Shift size	11.6 ms (256 pt)
Acoustic feature	log-mel spectrogram 80 dim
Encoder type	CNN + BLSTM
# char embedding dimension	512
# encoder layers	3 (CNN) & 1 (BLSTM)
# encoder BLSTM units	512
# encoder CNN filters	512
# BERT embedding dimension	1,024
# units in linear layer after BERT	512 (only for subword)
Encoder CNN filter size	5
Decoder type	LSTM
# decoder layers	2
# decoder units	1,024
# dimensions in attention	128
# filters in attention	32
Filter size in attention	31
# Prenet layers	2
# Prenet units	256
# Postnet layers	5
# Postnet filters	512
Postnet filter size	5
Dropout rate	0.5
Zoneout rate	0.1
λ in loss function	1.0 (for subword) 0.0 (for the others)
Learning rate	10^{-3}
Gradient clipping norm	1
Batch size	64
Epoch	200
Optimization method	Adam [38] w/ $\epsilon = 10^{-6}$

(a) *Mean Opinion Score (MOS)*: Subjects rated the naturalness of generated speech on a 5-point scale: 5 for excellent, 4 for good, 3 for fair, 2 for poor, and 1 for bad. Each subject rated 75 utterances, consisting of the output of all three models for 25 input text sequences.

(b) *A / B forced choice*: Subjects compare two audio samples corresponding to the same input text but generated by two different models and select their preferred choice, not necessarily on the basis of naturalness. For this evaluation, each subject rated 45 or 75 pairs.

3.3. Results

Experimental results for MOS are shown in Fig. 2. The proposed models improve the naturalness of generated speech, as represented by MOS, especially in the case of the subword-level model. In fact, the subword-level model gets a statistically significant improvement over both the baseline model ($p = 0.012$) and the phrase-level model ($p = 0.025$). These results suggest that utilizing text-context information aids in generation of more natural speech, although it is more useful at the finer granularity of subwords rather than at the coarser phrase level.

The experimental results for A / B forced choice are shown in Fig. 3. While as in MOS the subword-level model is preferable over the baseline model, the phrase-level model does worse than the baseline model.

On the basis of these two evaluation metrics, the subword-level model seems to outperform the baseline model. Interestingly, even though the BERT model is trained at the sentence level, it still shows its utility in representing the phrase-level utterances in LJSpeech. To get a better sense of cases where the scores for the two models diverge, we show examples in Fig. 4 where the raters have high agreement in their preference for one model or another. The subword-based model does worse on both of the examples containing proper nouns, which may be due to the choice of uncased BERT (which means it treats the proper noun phrases similarly to other words).

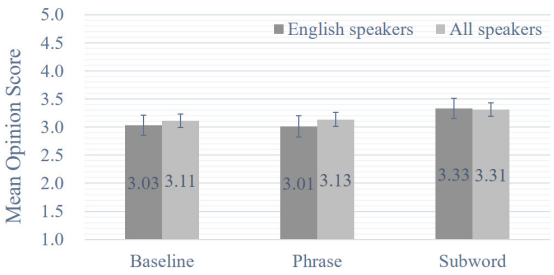


Figure 2: *Mean opinion scores for naturalness of the baseline, phrase-level, and subword-level models. The error bars represent 95 % confidence intervals.*

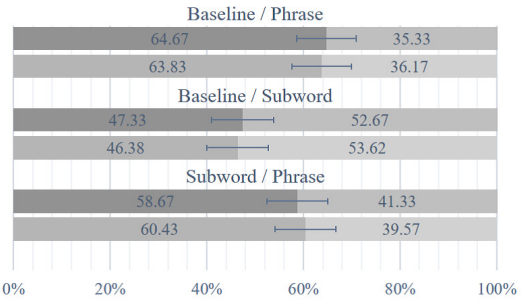


Figure 3: *Results of A / B forced choice. The error bars represent 95 % confidence intervals. The upper one represents the score of English speakers, and the lower one represents that of all speakers.*

050-0073: result in some degree of interference with the personal liberty of those involved.
050-0078: In June 1964, the Secret Service sent to a number of Federal law enforcement and intelligence agencies
050-0098: determination to use a means, other than legal or peaceful, to satisfy his grievance, end quote, within the meaning of the new criteria.
050-0046: it has obtained the services of outside consultants, such as the Rand Corporation,
050-0049: and from a specialist in psychiatric prognostication at Walter Reed Hospital.

Figure 4: *Examples from A / B forced choice with high agreement among listeners. The first three examples (in blue) are cases where the subword-based model is preferred, and the last two (in red) are ones where the baseline is preferred.*

4. Conclusion

We have proposed a TTS model that explicitly uses text-context information, in the form of pre-trained BERT text embeddings, and we have evaluated two variants of the model. Subjective listening tests using the LJSpeech database show that the subword-level model variant improves mean opinion scores modestly but significantly over a baseline TTS model without pre-trained text embedding input. This improvement is obtained in spite of the fact that we apply BERT, which is intended for full sentences, to a mix of different-sized phrases. It may be possible to obtain larger gains by applying the model only to complete sentences. Future work should further investigate the effect of this choice. In future work, we will also apply our approach to a wider variety of data sets, including various text domains and languages.

5. References

[1] P. Taylor, *Text-to-speech synthesis*. Cambridge University Press, 2009.

[2] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, 2009.

[3] S. Takamichi, K. Kobayashi, K. Tanaka, T. Toda, and S. Nakamura, "The NAIST text-to-speech system for the Blizzard Challenge 2015," in *Blizzard Challenge Workshop*, 2015.

[4] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *ICLR Workshop*, 2017.

[5] 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.

[6] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," in *ICLR*, 2018.

[7] 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 *ICASSP*, 2018.

[8] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech ," in *ICLR*, 2019.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *NIPS*, 2014.

[10] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional Sequence to Sequence Learning," in *ICML*, 2017.

[11] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-End Attention-based Large Vocabulary Speech Recognition," in *ICASSP*, 2016.

[12] 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:1609.03499*, 2016.

[13] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *ICML*, 2018.

[14] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A real-time speaker-dependent neural vocoder," in *ICASSP*, 2018.

[15] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient Neural Audio Synthesis," in *ICML*, 2018.

[16] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis," *arXiv:1811.00002*, 2018.

[17] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *ICASSP*, 2008.

[18] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, 2016.

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

[20] 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 *NIPS*, 2018.

[21] 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," in *ICML*, 2018.

[22] 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," in *ICML*, 2018.

[23] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *ICASSP*, 2019.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *NAACL*, 2019.

[25] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*, 2018.

[26] K. Ito, "The LJ speech dataset," 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NIPS*, 2017.

[28] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv:1609.08144*, 2016.

[29] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv:1607.06450*, 2016.

[30] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv:1606.08415*, 2016.

[31] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward Attention in Sequence-to-sequence Acoustic Modelling for Speech Synthesis," in *ICASSP*, 2018.

[32] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention," in *ICASSP*, 2018.

[33] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder." in *Interspeech*, 2017.

[34] G. Recommendation, "Pulse code modulation (PCM) of voice frequencies," *ITU*, 1988.

[35] K. Tachibana, T. Toda, Y. Shiga, and H. Kawai, "An investigation of noise shaping with perceptual weighting for WaveNet-based speech generation," in *ICASSP*, 2018.

[36] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Interspeech*, 2018.

[37] A. Paszke, S. Gross, S. Chintala, and G. Chanan, "PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration." 2017. [Online]. Available: https://github.com/pytorch/pytorch

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.