Journal of Artificial Intelligence Practice (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/jaip.2023.060109
ISSN 2371-8412 Vol. 6 Num. 1

Tibetan Lhasa Dialect Speech Synthesis Method Based on End-to-End Model

Zhihao Song, Guanyu Li[*], Guangming Li

Northwest Minzu University, Lanzhou, Gansu, 730000, China
[*]Corresponding author

Keywords: Tibetan Lhasa, end-to-end, speech synthesis

Abstract: End-to-end speech synthesis is now the most popular technique for Tibetan speech synthesis. This paper explores the field of Tibetan Lhasa speech synthesis using the Tacotron2 and VITS frameworks, which are based on the end-to-end methodology. To address the problem of inaccurate and incomplete coverage in the phoneme dictionary, a method of synthesizing Tibetan characters is used for Tibetan speech synthesis. Different sequence methods are used for synthesis in the Tacotron2 model, and experimental data indicate that using Tibetan characters in the Tibetan Lhasa speech synthesis has good performance in this model. Last but not least, the Tibetan character synthesis method employs the VITS framework, yielding results that are perfect for speech synthesis. Since Tibetan letters have great application value in Tibetan speech synthesis, the use of them as text input in the fully end-to-end VITS synthesis framework merits further study and promotion.

1. Introduction

Speech synthesis is a key technique for human-machine voice interaction, and its main goal is to produce continuous speech that is extremely eloquent and natural-sounding. End-to-end speech synthesis technology is now able to anticipate speech spectra straight from text, which is then translated into audio using a neural vocoder, thanks to the quick development of deep learning [1]. The neural network acoustic model plus vocoder structure has currently gained popularity in speech synthesis technology and has advanced to the point where it can produce synthesized speech in English and Chinese research that is very similar to that of a human. Tibetan is a crucial part of the Chinese language system since it has distinct ethnic characteristics, and Tibetan speech synthesis has a big impact on the smart development of different Tibetan regions. Early studies on the Tibetan language mostly concentrated on statistical parameter speech synthesis, acoustic and prosodic analysis, and Tibetan text regularization. Tibetan speech synthesis began relatively late, but it developed from concatenative synthesis throughout time [2] additionally to statistical parametric speech synthesis [3] to the era of Tibetan speech synthesis using neural networks. These days, deep learning and neural network advancements have not only lowered the bar for Tibetan speech synthesis [4] but also enhanced its quality, making Tibetan speech synthesis a crucial duty in the processing of Tibetan language data.

Using the text-to-phoneme conversion approach, there are still issues with inaccurate and missing words in the speech synthesis of the Tibetan language's Lhasa dialect. This study examines the end-to-end Tibetan Lhasa dialect speech synthesis technology from the two aspects of text input unit selection and speech synthesis acoustic model selection in an effort to improve the naturalness, intelligibility, and clarity of the speech synthesis of the Lhasa dialect of Tibetan. This research compares the effects of phonemes and Tibetan characters on the performance of Tibetan Lhasa speech synthesis and analyzes the structural characteristics of Tibetan text in terms of choosing the input units for text. Two distinct generative models are contrasted while deciding on the acoustic model, including the Tacotron2 two-stage speech synthesis model [5] based on the fully end-to-end model VITS and the end-to-end acoustic model [6]. According to the experiment, utilizing Tibetan characters as input text improves synthesis performance in the Tacotron2 framework reasonably well, and using Tibetan characters in the VITS framework also produces an excellent Tibetan Lhasa voice synthesis effect.

2. End-to-end Neural Network Structure

2.1. Sequence to Sequence Model

In 2017, Google Brain unveiled the end-to-end voice synthesis framework Tacotron2. The model consists of two components: the first is a recurrent seq2seq-based feature prediction network with an attention mechanism for anticipating frame sequences of the mel spectrograms from input character sequences. The second component, a neural vocoder, produces time-domain waveform samples based on the anticipated frame sequences of the mel spectrograms. The time-domain waveform, which is smoother and easier to train using mean squared error loss (MSE), since each of its frames is phase-invariant, can be used to calculate the mel spectrograms, which are used to link the two components of the system. Figure 1 depicts Tacotron2's overall system architecture.

An encoder and a decoder that introduces an attention mechanism make up the front-end acoustic model. The input sequence is transformed by the encoder into an implicit representation sequence, which is given to the decoder in order to anticipate the voice spectrogram. Three convolution layers are used to convolve a 512-dimensional character vector created from the input characters, each layer containing 512 5×1 convolution kernels, i.e., each kernel spanning 5 characters, followed by batch normalization [7] and the activation of ReLU. The input character sequence's broad context is modeled by the convolutional layers. A Bi-LSTM layer receives the output of the final convolutional layer and produces encoded features with 512 units.

When the encoder outputs are combined using the attention network, the attention network summaries each encoded sequence into a fixed-length context vector [8]. The model can stay consistent as it moves forward along the input sequence, minimizing potential decoding processing errors like subsequence duplication or omission. This is accomplished by using a position-sensitive attention mechanism, which expands the additive attention mechanism to enable the use of cumulative attention weights from earlier decoding processes as additional features. Following the computation of the attention weights and the transfer of the input sequence and location features to a 128-dimensional hidden layer representation, the location features are obtained by convolving 32 1-dimensional convolutional kernels of length 31.

The decoder, which predicts the output spectrogram from the encoded input sequence one frame at a time, is an autoregressive recurrent neural network. First, a two-layer fully linked Pre-Net with 256 hidden ReLU cells per layer receives the spectrum frames predicted in the preceding phase. A two-layer stacked one-way LSTM with 1024 cells receives the output of the Pre-Net stitched with the attention context vector. The attention context vector is once more used to stitch the output of the LSTM together before undergoing the desired spectral frame is predicted using a linear transformation. Each layer of the Post-Net consists of 512 5×1 convolution kernels and a batch normalization process, every batch normalization procedure is followed by a tanh activation function before the neural coder WaveGlow, with the exception of the final layer of convolution. The neural coder WaveGlow creates the audio [9].

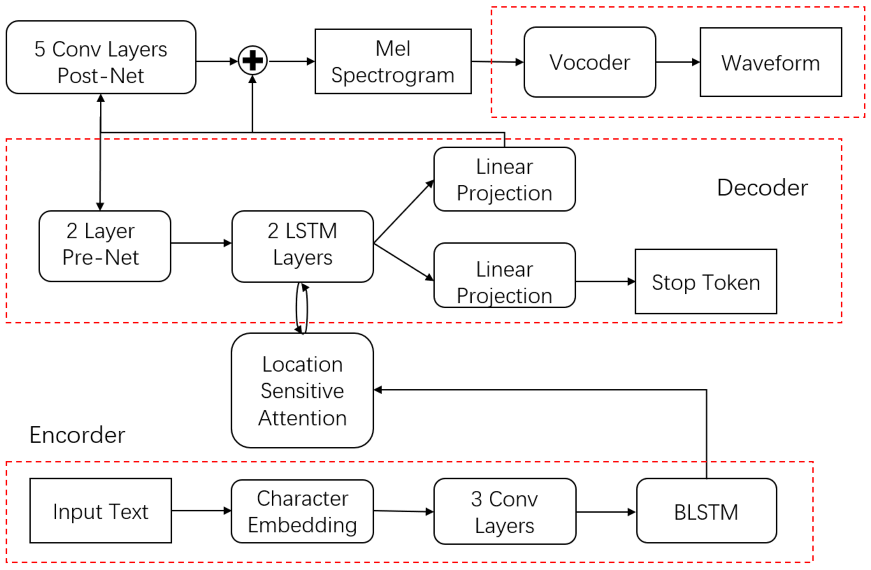

Figure 1: Tacotron 2 system framework diagram

2.2. Full End-to-End Model

The core component of the VITS speech synthesis model, which is fully end-to-end, is variational inference, together with normalized flow and adversarial training. VITS travels through the hidden variables, models randomly on the hidden variables, and employs a stochastic duration predictor to increase the expressiveness of the synthesized speech, in contrast to Tacotron2, which uses a spectrally connected acoustic model and vocoder. A posterior encoder, a projection encoder, a decoder, a discriminator, and a stochastic duration predictor make up the model's overall structure. The inference synthesis stage does not use the posterior encoder and discriminator; they are only used for training. Figure 2 depicts a block schematic of the complete VITS system, with the solid line representing the training phase process and the dotted line representing the inference process.

For the posterior encoder, the non-causal WaveNet residual module [10] used in WaveGlow and Glow-TTS [11] is used. An extended convolutional layer with gated activation units and skipped connections makes up the residual module. The posterior distribution is a typical Gaussian distribution, and a linear layer is used to construct the mean and variance of the normal posterior distribution.

The prior encoder contains two parts, consisting of a text encoder and a normalized flowf_θ. The text encoder processes the input sequence, processing the text input sequence such as phoneme or Tibetan characters sequence as c_{text}, which uses the encoder module of the transformer [12], which uses relative position representation instead of absolute position encoding [13], by which the hidden representation h_{text} can be obtained from c_{text}, and finally passes through the linear layer to generate the mean and variance. The standardized flow is a collection of affine coupling layers used to increase the prior distribution's flexibility and transform it into a more complicated distribution [14].

The HiFi-GAN V1 generator is used to produce the decoder. A multi-receiver field fusion module follows each transposed convolution in the decoder. The aggregate of the outputs from the remaining blocks with various receiver field widths constitutes the multi-receiver field fusion module's output. The decoder is used to create audio and functions similarly to the vocoder.

The discriminator makes use of the HiFi-GAN multi-period discriminator architecture [15]. A hybrid Markovian window-based sub-discriminator called the multi-period discriminator [16], each of them processes the incoming waveform's period pattern differently.

Using the input text, the stochastic duration predictor creates a distribution of text durations. The stochastic duration predictor is effectively reparameterized by superimposing the residual blocks with dilated and depth-separable convolutional layers. The coupling layer, which consists of a reversible nonlinear transformation of monotone rational quadratic splines, is then subjected to the neural spline flow. Neural spline flows [17] compared to often employed affine coupling layers, increase transformation expressiveness with a similar amount of parameters.

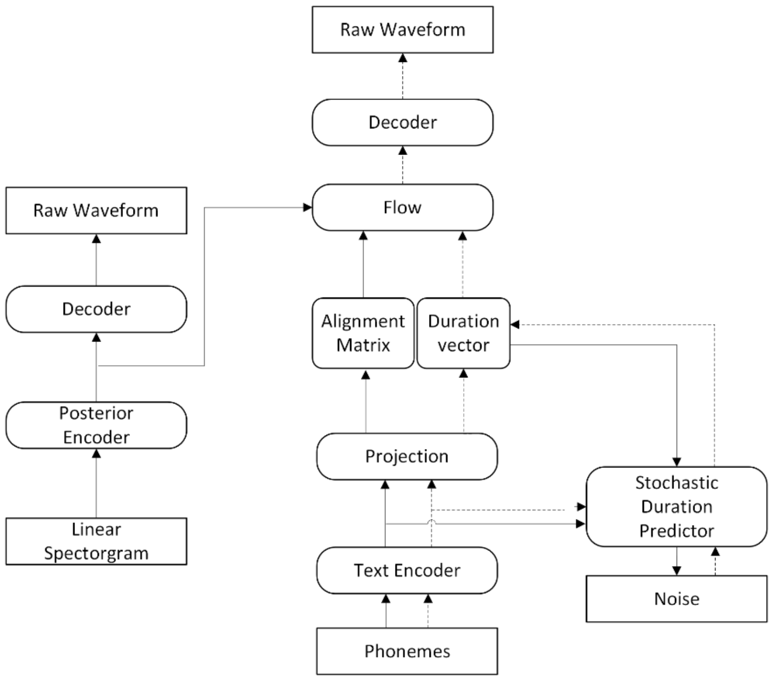

Figure 2: VITS system framework diagram

3. Experiments and Results

This experiment uses the Tacotron2 framework and VITS framework to implement the speech synthesis of the Tibetan Lhasa language. The detailed experimental procedure is as follows.

3.1. Experimental Data and Processing

The experimental corpus was adopted from the internal speech synthesis corpus of the speech engineering experiment of Northwest Minzu University, including 9387 sentences of professional Tibetan Lhasa male voice, with a sampling rate of 22050 Hz, sampling accuracy of 16 bits, single channel, and a total duration of 9.19 hours. The Tibetan corpus of the Lhasa dialect was selected to cover various linguistic features as much as possible, evenly covering various phonological phenomena, and at the same time designed according to the linguistic knowledge of Tibetan, including all the consonants and vowels in Tibetan, which can reduce the problem of poor generalization of the model due to the lack of training data or sparse data, and the text covers a wide range of aspects, including ethnic customs, daily life, music and art, etc. content. Finally, Voice Activity Detection (VAD) processing is performed on the silences of the speech in the corpus. Silences play a crucial role in the neural network speech synthesis system, and too many silences will slow down the learning of the model using the attention mechanism, and removing the silences at the beginning and end of the speech can speed up the convergence of the model.

On the self-built pronunciation dictionary, the phonetic symbols corresponding to the syllables of Tibetan Lhasa were converted into a list of phonemes, thus converting the set of Tibetan phonemes into English alphabetic representations, for example, the Tibetan word " ཐད་དེ་ཁོ་ར་ཨ་མའི་འགྲམ་སྔོར་འགྲོ་གི་རེད། ", was transformed into " kh o r a i kh a ng s ew ds u m a ds u t i ".

For the speech synthesis of Tibetan Lhasa using Tibetan characters as input text, the approach taken was to first count all the alphabetic Unicode codes of Tibetan characters appearing in the corpus, from 0F40-0FB7, and to count a total of 58 letters after removing the special symbols from them. All single characters and Tibetan stacked combination symbols, such as the base character, head letter, subjoined letter, and vowel, are referred to as Tibetan characters. For Tibetan, the syllable point "་" means to divide the syllable, so it is used to divide the word, and the clause line "།" means to convert the comma and the stop to ",".

3.2. Experimental Environment and Parameter Settings

The environment required for this experiment is as follows. The implementation is done on a centos 7 system using python, with a Tesla T4 as the graphics card.

Tacotron2's short-time Fourier transform (STFT) with a frame size of 50 ms, frame hop of 12.5 ms, and Hann window function is used to produce the mel spectrograms. Using an 80-channel mel filterbank with a frequency range of 125 Hz to 7.6 kHz, the STFT magnitude is converted to the mel scale before being subjected to log dynamic range compression. The output magnitudes of the filterbank are trimmed to a minimum value of 0.01 before log compression in order to reduce dynamic range in the logarithmic domain. The batch size is configured to 64. We employ the Adam optimizer with β_1=0.9, β_2=0.999， ϵ=10^{-6}, and a learning rate of 10^{-3} that exponentially decays to 10^{-5} starting after 50,000 iterations.

Model networks in the VITS framework are trained using the AdamW optimizer with β_1=0.8， β_2=0.99, and weight decay λ=0.01. The learning rate decay is scheduled by a $0.999^{1/8}$ factor in every epoch with an initial learning rate of 2×10^{-4}. The batch size is set to 64 and a window generator is utilized for training to cut down on training time and memory utilization.

3.3. Experimental Results

To more accurately reflect the subjective perception of speech synthesis quality, the mean opinion score (MOS) evaluation is applied. In this experiment, we got 20 native speakers from the Lhasa region of Tibet to help us score and then trained these people to score the 20 speech items for each model. The following MOS scoring results using Tibetan Lhasa phonemes as input in the Tacotron2 framework, Tibetan characters as input in the Tacotron2 framework, and Tibetan characters in the VITS framework are shown in Table 1:

Table 1: Mean Opinion Score (MOS) scoring results

System	MOS
Tacotron2+WaveGlow (phoneme)	3.93
Tacotron2+WaveGlow (Tibetan characters)	4.08
VITS (Tibetan characters)	4.20
Ground truth	4.48

The synthesis impact of the aforementioned models was compared using the same Tibetan sentence of the above models is used: "ཉལ་ཁང་ལ་འདྲས་བཤད་ཀ་རྩ་ཅི་བྱེད་ཀི་རེ་པ།མཚམས་ཅིག་སྐབས་དུས། "The mel spectrogram is shown below:

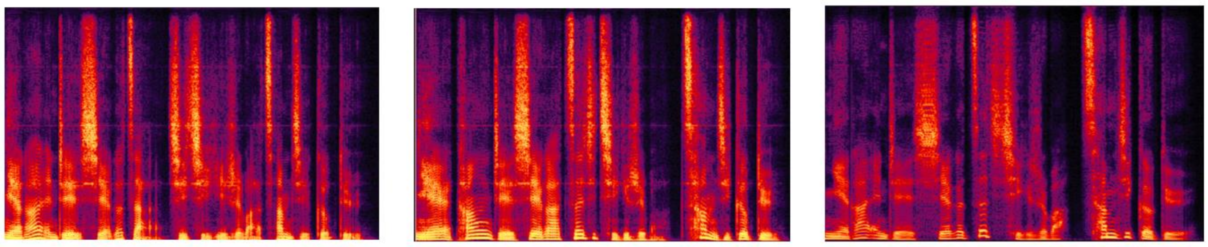

Figure 3: Different models use different sequences to generate the mel spectrograms of the same Tibetan sentence

According to Figure 3, which contrasts the left and right graphs, the model for Tibetan Lhasa speech synthesis using the Tibetan characters will perform better because the resonance peaks are sharper and more distinct after using the Tibetan characters sequence as input than the model using the phoneme sequence, and the high-frequency region is described in more detail. Finally, observe the middle and right graphs, the right graph is also cleaner and clearer in the low frequencies, the high-frequency region in the middle frequencies does not have the continuous horizontal lines in the middle left graph, and the synthesized voice is the less mechanical sound of the Tacotron2 model and closer to the original voice.

4. Conclusion and Future Work

The VITS and Tacotron2 models are established models in the field of voice synthesis technology at this time. When using the Tibetan character sequence approach, both objective and subjective testing have shown that these models exhibit good performance in Tibetan Lhasa speech synthesis. This study conducts a number of trials using cutting-edge technology and the distinctive Tibetan language. The results can serve as guidelines and pointers for future studies in Tibetan speech synthesis, furthering the field's advancement and ethnic harmony.

In future research, it is hoped that the models can be further optimized through adjustments to improve training and that existing models can be used for transfer learning of the small data set of the Tibetan Amdo dialect to achieve comparable synthesis results.

Acknowledgments

This research was financially supported by a Basic scientific research business project of central universities (31920220010).

References

[1] Xu Tan et al. “A Survey on Neural Speech Synthesis.” arXiv: Audio and Speech Processing (2021): 15-16.
[2] Rangzhuoma C., and C. Zhijie. "Unit Selection Algorism for Corpus-based Tibetan Speech Synthesis." Journal of Chinese Information Processing 31.5 (2017): 59-63.
[3] Zhou Y., and D. C. Zhao. "Research on HMM-based Tibetan speech synthesis." Computer Applications and Software 32.5 (2015): 171-174.
[4] Zhao Yue, et al. "Lhasa-Tibetan speech synthesis using end-to-end model." IEEE Access 7 (2019): 24-30.
[5] Jonathan Shen et al. “Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions” international conference on acoustics, speech, and signal processing (2017): 56-60.
[6] Jaehyeon Kim et al. “Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech” international conference on machine learning (2021): 42-44.
[7] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift” international conference on machine learning (2015): 57-60.

[8] Jan Chorowski et al. "Attention-Based Models for Speech Recognition" neural information processing systems (2015): 11-15.
[9] Ryan Prenger et al. "Waveglow: A Flow-based Generative Network for Speech Synthesis" international conference on acoustics speech and signal processing (2019): 57-70.
[10] Aaron van den Oord et al. "WaveNet: A Generative Model for Raw Audio" arXiv: Sound (2016): 142.
[11] Jaehyeon Kim et al. "Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search" neural information processing systems (2020): 1-10.
[12] Ashish Vaswani et al. "Attention Is All You Need" (2022): 15-19.
[13] Peter Shaw et al. "Self-Attention with Relative Position Representations" North American chapter of the association for computational linguistics (2018): 67-70.
[14] Laurent Dinh et al. "Density estimation using Real NVP" Learning (2016): 75-77.
[15] Jungil Kong et al. "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis" neural information processing systems (2020): 80-82.
[16] Kundan Kumar et al. "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis" neural information processing systems (2019): 90-94.
[17] Conor Durkan et al. "Neural Spline Flows" neural information processing systems (2019): 95-100.