IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 4, 2020, accepted September 20, 2020, date of publication September 29, 2020,
date of current version October 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027619

Hierarchical Transfer Learning for Multilingual, Multi-Speaker, and Style Transfer DNN-Based TTS on Low-Resource Languages

KURNIAWATI AZIZAH, (Member, IEEE), MIRNA ADRIANI, (Member, IEEE), AND WISNU JATMIKO, (Senior Member, IEEE)
Faculty of Computer Science, Universitas Indonesia, Depok 16424, Indonesia

Corresponding authors: Kurniawati Azizah (kurniawati.azizah@cs.ui.ac.id) and Wisnu Jatmiko (wisnuj@cs.ui.ac.id)

This work was supported in part by the Universitas Indonesia through the Publikasi Terindeks International (PUTI) Doktor under Grant NKB-566/UN2.RST/HKP.05.00/2020, and in part by the NVIDIA DGX-1 Computing Facilities at the Tokopedia-UI AI Center of Excellence.

ABSTRACT This work applies a hierarchical transfer learning to implement deep neural network (DNN)-based multilingual text-to-speech (TTS) for low-resource languages. DNN-based system typically requires a large amount of training data. In recent years, while DNN-based TTS has made remarkable results for high-resource languages, it still suffers from a data scarcity problem for low-resource languages. In this article, we propose a multi-stage transfer learning strategy to train our TTS model for low-resource languages. We make use of a high-resource language and a joint multilingual dataset of low-resource languages. A pre-trained monolingual TTS on the high-resource language is fine-tuned on the low-resource language using the same model architecture. Then, we apply partial network-based transfer learning from the pre-trained monolingual TTS to a multilingual TTS and finally from the pre-trained multilingual TTS to a multilingual with style transfer TTS. Our experiment on Indonesian, Javanese, and Sundanese languages show adequate quality of synthesized speech. The evaluation of our multilingual TTS reaches a mean opinion score (MOS) of 4.35 for Indonesian (ground truth = 4.36). Whereas for Javanese and Sundanese it reaches a MOS of 4.20 (ground truth = 4.38) and 4.28 (ground truth = 4.20), respectively. For parallel style transfer evaluation, our TTS model reaches an F0 frame error (FFE) of 9.08%, 10.13%, and 8.43% for Indonesian, Javanese, and Sundanese, respectively. The results indicate that the proposed strategy can be effectively applied to the low-resource languages target domain. With a small amount of training data, our models are able to learn step by step from a smaller TTS network to larger networks, produce intelligible speech approaching the real human voice, and successfully transfer speaking style from a reference audio.

INDEX TERMS Deep neural network, hierarchical transfer learning, low-resource, multi-speaker, multilingual, style transfer, text-to-speech.

I. INTRODUCTION

Speech is the most natural verbal communication tool that can be easily understood by normal humans [1]. The computer's ability to process voice signals is necessary in the area of human computer interaction (HCI). It helps the computer to communicate and interact with humans or to be used as a communication device between normal humans and visual/speech impaired people. Text-to-speech (TTS) learns how a computer can read text or symbols and pronounce them by producing sound waves automatically [2]. The purpose of building TTS is to produce synthesized speech that can be easily understood and is indistinguishable from sound produced by real humans [3]. In general, modern TTS involves three main processes: text analysis, acoustic modeling, and synthesizing speech waveforms [4]. Model-based TTS research has been dominated by statistical parametric speech synthesis (SPSS) [5]–[9] until recent years in which deep learning has delivered extraordinary achievements in various fields [10]–[12]. This has attracted

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

many researchers to exploit DNN in all TTS process stages. Beyond parametric speech synthesis (BPSS) that applies DNN for both features and rules learning has become increasingly researched [13]–[22]. Tacotron-2 [13], a state-of-the-art DNN-based TTS that can be trained end-to-end using <text, audio> data pairs, successfully produces human-like synthesized speech. Besides producing high-quality synthetic sounds, DNN-based TTS introduces many possibilities to produce speech in various types of sounds, speech styles, and emotional states. E2E-prosody [23], Tacotron-GST [24], and Mellotron [25] proposed a DNN-based prosody model that has an important role in transferring a reference speaking style when generating synthesized speech. These works showed satisfactory results.

Despite the remarkable performance, DNN-based TTS has a very strong dependence on a large amount of training data to understand latent data patterns. This data dependency is one of the most serious problems for DNN-based model. The scale of the DNN model and the size of required training data correlate almost linearly [26]. Based on our preliminary study, a minimum 10 hours of data is required to train Tacotron-2-based single-speaker monolingual TTS on the Indonesian domain. Training data below 3 hours is unable to produce intelligible speech. As for multi-speaker multilingual TTS, 10 hours of data is still insufficient to train the model. This study confirms that the bigger the scale of the TTS network, the bigger the amount of training data needed. Rather than building a bigger dataset that is expensive and needs human efforts, it is necessary to find alternative strategies to train the model on low-resource language domains.

There have been several efforts to train a DNN-based TTS model using a small amount of annotated <text, audio> data pairs. Semi-supervised training proposed by [27] to make use of textual and acoustic knowledge from non-parallel large text and speech corpora for training end-to-end TTS with a small amount of parallel data. Other studies used cycle consistency training using the automatic speech recognition (ASR) model to train TTS [28], [29]. In the training process, ASR is used to look for transcripts from sounds, while TTS reconstructs transcripts into sounds. However, these approaches still require a large amount of unlabelled text and unlabelled audio corpora that are limitedly available for low-resource domain. Speech chain machine for cross-lingual is proposed by [30] that applies cycle consistency training for cross-lingual ASR-TTS. Work [31] proposed an approach to discover cross-lingual symbol mapping from abundant source data.

Transfer learning is an interesting option to overcome the lack of data in low-resource language by allowing what has been learned in a source domain be exploited to improve generalization in a target domain. Referring to the classification of transfer learning approaches in traditional machine learning by [32], [33], there are four transfer categories: instance transfer, feature-representation transfer, parameter transfer, and relational-knowledge transfer. Especially for deep learning, a study by [26] classifies deep transfer learning (DTL) into different four categories: instances-based DTL, mapping-based DTL, network-based DTL, and adversarial-based DTL. Some DNN-based systems have successfully applied deep transfer learning, including TTS [31], image classification [34], [35], machine translation [36], [37], automatic speech recognition [38]–[40], language identification [41], and sentiment classification [42].

We propose hierarchical transfer learning, a network-based DTL, to train the TTS model on low-resource (target) languages by utilizing a high-resource (source) language. This strategy is a multi-stage learning inspired by the human learning process to accumulate knowledge from previous learning, step by step from a simple task to more complex ones. Furthermore, we exploit the benefit of using a joint multilingual dataset of low-resource languages to maximize the latent variable learning from more data of other languages. For this reason, we develop DNN-based multilingual multi-speaker TTS with and without style transferring by extending the Tacotron-2 architecture with additional networks for multi-speaker, multilingual, and style transfer. Adding a multilingual component has two benefits. First, TTS model can learn from more data of other languages. More data can generalize the network parameters better. Second, it allows a native speaker of a language to speak fluently in other languages. We train TTS models using the proposed hierarchical transfer learning in several stages. For each transfer stage, it has a background motive to transfer particular knowledge from previous learning: parameter generalization including alignment map between text input and spectrogram output from a high-resource language, pronunciation learning from a phonologically close language, and multilingual multi-speaker learning from a joint multilingual data. After these learned capabilities are transferred, the TTS model at the last stage learns to imitate the speaking style from a reference audio.

Our experiment uses an English dataset as the source domain and a joint multilingual dataset of Indonesian, Javanese, and Sundanese as the target domain. These target languages are phonologically close. Using international phonetic alphabet (IPA), Indonesian has 32 phonemes, while the other languages have all Indonesian phonemes with additional three phonemes for Javanese and one phoneme for Sundanese. The models are able to generate synthesized speech that is close to a real human voice by training them using less than 1 hour of monolingual dataset and 11 hours of joint multilingual dataset (Javanese and Sundanese are less than three hours each). Our study reports that these amounts are inadequate to train the TTS models from scratch. In comparison, single-speaker monolingual Tacotron-2 uses more than 24 hours, multi-speaker monolingual Mellotron uses 44 hours and 41.7 hours, and monolingual E2E-Prosody uses 147 hours and 296 hours for single-speaker and multi-speaker, respectively. TTS evaluation for Indonesian and Sundanese reaches a smaller MOS difference from the real human speech than the baseline Tacotron-2 on English and better mel-cepstral distortion (MCD) than the baseline

E2E-Prosody on English. As for transfer style, our model on female speakers provides better FFE than Mellotron.

In summary, our main contributions are as follows:

1. We present Tacotron-2-based TTS that supports multi-speaker, multilingual, and style transfer by adding new network components. Multilingual component enables TTS model to be trained on a joint multilingual dataset. The joint dataset can help TTS model improve the generalization learning of a low-resource language using more data from other languages with phonetic similarity and allow a speaker of one language to speak fluently in other languages with/without style transfer.
2. We propose a hierarchical transfer learning scheme to train TTS for low-resource languages in several stages. Firstly, it utilizes pre-trained model on a high-resource single-speaker monolingual source domain and fine-tune on a single-speaker monolingual target domain. Secondly, we use a partial network-based DTL from the pre-trained single-speaker monolingual TTS to build a multi-speaker multilingual TTS that is fine-tuned using a joint multilingual dataset. Finally, similar partial network-based DTL is used to build a multi-speaker multilingual with style transfer TTS from the pre-trained multi-speaker multilingual TTS model.

The rest of the paper is organized as follows: Section II presents previous related works. Section III introduces our DNN-based TTS architecture and proposed hierarchical transfer learning. Section IV provides implementation details. Section V presents the experimental results and Section VI concludes the study.

II. RELATED WORKS

A. END-TO-END DNN-BASED TTS

A recent promising beyond parametric speech synthesis (BPSS) is the end-to-end TTS system that combines the main stages of the TTS process into a DNN framework that can be trained directly using <text, audio> data pairs. There are several advantages of such an integrated end-to-end TTS system [4]: It does not require phoneme level alignment and reduces the need for exhausting engineering features; It is easier for conditioning on various attributes, such as speakers, languages, or high-level features such as sentiment; It is easier to adapt to new data; It tends to be stronger than a multi-stage model where the errors of each component can accumulate. Tacotron-2 [13], a simplification of Tacotron [15], is a fully end-to-end DNN-based TTS system that can be trained directly using <text, audio> data pairs and directly processes raw orthographic text to produce spectograms. Tacotron-2 uses WaveNet [16] as a vocoder conditioned on the mel-spectogram instead of using the Griffin-Lim algorithm as in Tacotron.

To convey human-like speech, the TTS system needs to learn how to make a prosody model, such as paralinguistic information (intention, attitude, and emotion), pitch, rhythm, intonation, stress, and style. Tacotron [15] and Tacotron-2 [13] do not model prosody explicitly. E2E Prosody [23] added a reference encoder network to Tacotron architecture as a prosodic modeling derived from a reference audio. Tacotron-GST [24] proposed modeling speech style using global style token (GST) by adding style token layer that consumes the reference encoder outputs [23] using a multi-head attention scheme [43]. Recently, Mellotron [25] combined GST, pitch, and rhythm for style transferring and successfully reduced F0 frame error (FFE) significantly between synthesized audio and reference audio.

Different from these end-to-end TTS models, we add multilingual component to exploit the benefits of using a joint multilingual dataset. We also extend Tacotron-2 to support style transfer using GST [24] by conditioning the decoder with pitch and rhythm obtained from a reference audio signal as applied in Mellotron. GST can express various expressive styles without requiring explicit prosody labels. The GST network is jointly trained with the whole model that is only driven by the reconstruction loss of the Tacotron-2 decoder. However, unlike Mellotron that uses phoneme-level in the text processing, our approach uses character level. Therefore, it does not need to make a phonetic dictionary that requires human annotation effort. As for the vocoder, we employs WaveGlow [44] instead of WaveNet used by Tacotron-2. Unlike WaveNet that produces very natural speech waveforms but is very slow due to the autoregressive generation process, WaveGlow is a non-autoregressive vocoder that provides fast, efficient and high-quality audio synthesis without auto-regression.

B. LOW-RESOURCE PROBLEM

Deep learning has a very strong dependence on a large amount of training data. Previous studies related to data efficiency for training the DNN-based TTS model [27]–[29] are less suitable for low-resource language. Even though these approaches do not require a large amount of parallel data, they still need a large amount of non-parallel text and speech corpora. ASR-TTS proposed by [30] need additional ASR to assist TTS learning. Mapping-based DTL is explored in [31] by adding a phonetic transformation network (PTN) model to learn a mapping between source and target linguistic symbols. An ASR system is used to train PTN separately. However, this approach can only be applied to the same TTS network. It does not have the flexibility to transfer the learning on a more complex network.

Different from the solutions proposed in [27]–[29], our proposed strategy does not require a large amount of non-parallel text and audio corpora. Our strategy is simpler than [30] as it does not need additional system such as ASR. Similar to [31], we apply DTL approach. However, our DTL is network-based approach that is more flexible than mapping-based DTL applied by [31] in which with multi stages of transfer learning the previous learned DNN parameters can be passed on to a larger network. The hierarchical transfer learning scheme proposed in this article is an extended study of our previous work [45]. This prior

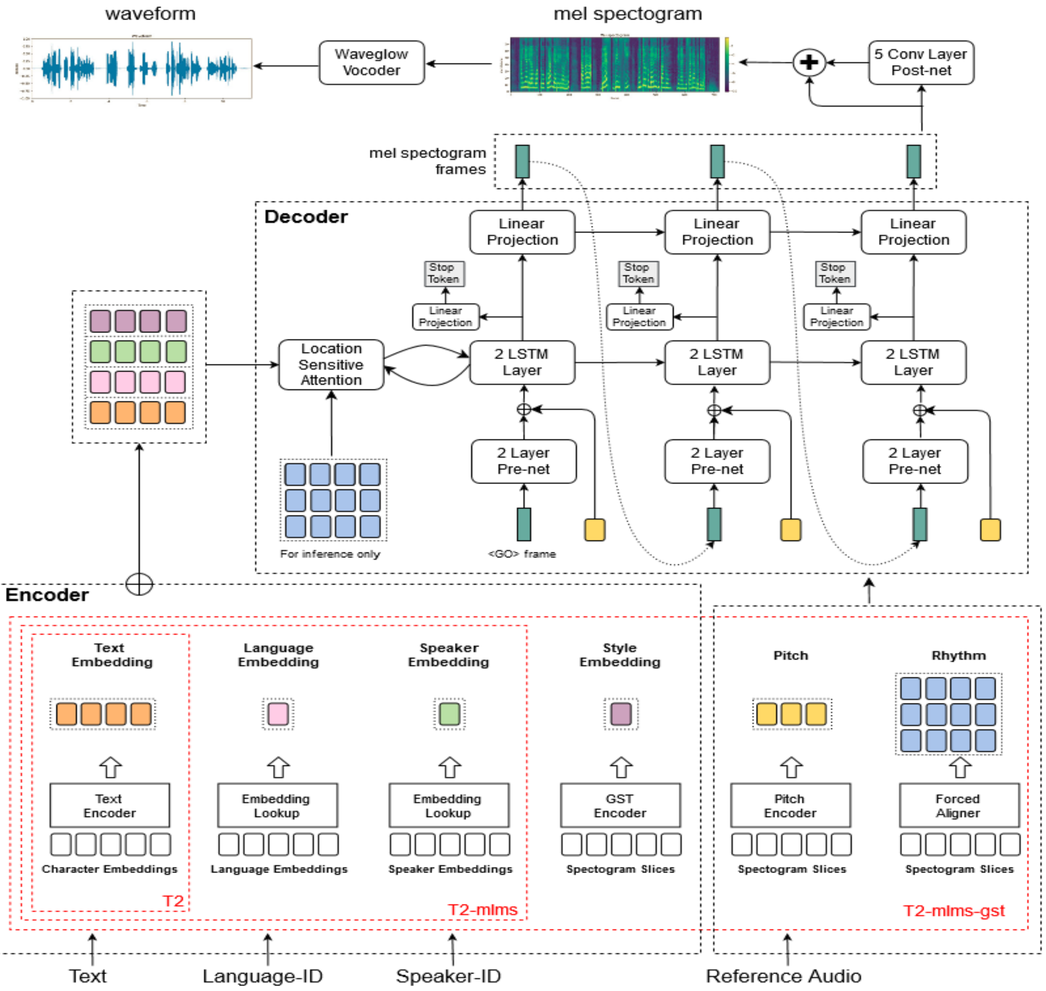


FIGURE 1. **TTS Architecture. The prediction network of T2-mlms contains T2, whereas T2-mlms-gst prediction network contains T2-mlms such that T2 ⊂ T2-mlms ⊂ T2-mlms-gst. All models use WaveGlow as a vocoder.**

work used pre-trained TTS model on a source domain and fine-tune it on a target domain using the same monolingual single-speaker TTS model. Our new proposed transfer scheme can be applied to both the same TTS model trained on a monolingual target domain and different, more complex models trained on a joint multilingual target domain. The new scheme exploits the benefit of generalization learning from other languages with phonetic similarity, allows a speaker of a language to speak other languages, and transfers speaking style from one speaker to another speaker.

III. METHODS

This section explains the proposed TTS architecture and hierarchical transfer learning training strategy.

A. MODEL ARCHITECTURES

TTS architecture in our work consists of three modules: Encoder module that converts inputs into feature representations; Decoder module that changes the representation of features into the acoustic parameters mel-spectogram; Vocoder module that produces sound signals from mel-spectogram. The encoder-decoder network can also be called spectogram prediction network that predicts spectrogram output from text input.

The entire proposed multilingual multi-speaker TTS model, illustrated in Figure 1, is a sequence-to-sequence (seq-to-seq) Tacotron-2 network [13] with some additions: style embedding as in [24], pitch contour and attention map as in [25], language embedding, and speaker embedding. These additional networks are for handling multilingual, multi-speaker, and transfer of speaking style, pitch, and rhythm from a reference audio.

There are three TTS models used in our study: T2, a Tacotron-2-based encoder decoder architecture; T2-mlms, an extension of T2 by adding language embedding and speaker embedding; T2-mlms-gst, an extension of T2-mlms with the addition of GST encoder, pitch, and rhythm components for prosody transferring.

1) TEXT ENCODER

Text encoder generates a $T_X\_d_X$-dimensional representation of the grapheme sequence. T_X is the length of the encoded text (usually the same as the transcript length) and d_X is the dimension of the encoded text. We adopt the text encoder networks used in Tacotron-2. It consists of learnable d_X-dimensional character embedding, followed by stacked convolutional layers with filter that spans 5 characters to model long-term context (N-grams) of the input sequence. After batch normalization and ReLU activation, the output of the convolutional layer is passed into a single bi-LSTM with d_X units.

2) MULTILINGUAL MULTI-SPEAKER

To model multilingual and multi-speaker, we use learnable d_L-dimensional language embedding network and d_S-dimensional speaker embedding network that are jointly trained with the TTS task without the need for changes in loss metrics. The training process updates the parameters so that similar languages/speakers in relation to synthesis task have close distance in the vector space. For both multilingual and multi-speaker model, we use a channel-wise embedding concatenated with the encoder output.

3) GLOBAL STYLE TOKEN

To model acoustic expressiveness we apply a style embedding using a GST encoder to capture speaking style from a reference audio as in [24]. The GST encoder calculates a d_G-dimensional style embedding that corresponds to the mel-spectogram of a reference audio. It consists of reference encoder network as in [23] followed by a style token layer that are jointly trained with the rest of the model, driven by the reconstruction loss from TTS decoder.

4) PITCH CONTOUR

GST only offers rough control over expressive speech characteristics. To carry out finer and detailed control, we add networks to condition melodic information such as pitch and rhythm. In addition to GST network, we adopt scheme in [25] to explicitly model expressive speech variables, such as fundamental frequency contour (F0) or pitch, and voicing decision (voiced/unvoiced), and rhythm variables. The pitch contour is extracted using the YIN algorithm [46] with a harmony threshold between 0.1 and 0.25 from the reference audio. The pitch goes to a convolutional layer followed by ReLU to get d_P-dimensional pitch representation.

5) RHYTHM

Rhythm, also called alignment map, is learnt from text and spectrogram as described in [13] by using location-sensitive attention [47], which is an extension of additive attention [48]. Alignment map is a $T_M\_T_X$-dimensional matrix that contains alignment (or attention weight) of an input text X with the length of T_X characters and reference mel-spectogram M with the length of T_M frames. By learning the alignment map during training, we can control the rhythm during inference. Alignment map is extracted using a forced-aligner from <reference audio, transcription> pair data, as in [25]. TTS can produce the same rhythm as the reference audio using the extracted alignment map.

6) WAVEGLOW

WaveGlow is a non-autoregressive vocoder that is able to convert mel-spectrogram into waveforms faster than real time [44]. WaveGlow combines flow-based generative model Glow [49] and WaveNet [16] to achieve the generation of non-autoregressive waveforms, making it possible to speed up the training process on a large scale while maintaining the naturalness of synthesized speech. WaveGlow vocoder consists of a single network that is trained using a single cost function to maximize the likelihood of training data and make training procedures simpler and more stable.

B. PREDICTION NETWORK FORMULATION

The following section describes the spectrogram prediction network formulation for T2, T2-mlms, and T2-mlms-gst in more detail. The spectogram prediction network is the encoder and decoder part of the architecture illustrated in Figure 1. It is a seq-to-seq model that converts an input text sequence $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{Tx})$ into an output spectrogram sequence $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T_Y})$. Each $\boldsymbol{y}_t$ is predicted based on all previous outputs $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1}$. The prediction is computed using the attention-based encoder decoder scheme.

1) MODEL T2

The proposed T2 model adopts the spectrogram prediction network used by Tacotron-2 [13]. In T2 model, the encoder processes input text sequence $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{Tx})$, where T_X is the number of characters in the text that has been normalized, and then converts them into $T_X\_d_X$-dimensional hidden representations $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx})$ in the following way:

$$H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx}) = encoder_{\theta_e}(X), \tag{1}$$

where θ_e is the encoder model parameters. The hidden representations $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx})$ are processed by the decoder network to produce predicted mel-spectrogram $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{TY})$ from which the vocoder generates speech waveforms. To produce output $\boldsymbol{y}_t$, the decoder calculates a new decoder hidden state $\boldsymbol{s}_t$ based on the prior state $\boldsymbol{s}_{t-1}$, prior output $\boldsymbol{y}_{t-1}$, and attention context vector $\boldsymbol{c}_t$. The decoder state $\boldsymbol{s}_t$ is formulated as follows:

$$\boldsymbol{s}_t = decoder_{\theta_d}\left(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{c}_t\right), \tag{2}$$

where θ_d is the decoder model parameters, and $\boldsymbol{c}_t$ is the context vector and is computed using attention scheme:

$$\boldsymbol{c}_t = \sum_i^{T_X} \alpha_{t,i}\boldsymbol{h}_i, \tag{3}$$

where $\alpha_{t,i}$ is the attention weight and is calculated as follows:

$$\alpha_{t,i} = softmax\left(\boldsymbol{e}_{t,i}\right), \tag{4}$$

where $\boldsymbol{e}_{t,i}$ is the attention score or energy that is calculated using location-sensitive attention as follows:

$$\boldsymbol{e}_{t,i} = \boldsymbol{w}^T tanh\left(\boldsymbol{W}\boldsymbol{s}_{t-1} + \boldsymbol{V}\boldsymbol{h}_i + \boldsymbol{U}\boldsymbol{f}_{t,i} + \boldsymbol{b}\right), \tag{5}$$

$$\boldsymbol{f}_t = \boldsymbol{F} * \alpha_{t-1}, \tag{6}$$

where $\boldsymbol{s}_{t-1}$ is the decoder hidden state from the prior time step, $\boldsymbol{h}_i$ is the i^{th} encoder hidden state, $\boldsymbol{f}_{t,i}$ is the location feature ($*$ is a 1-dimensional convolution operator). $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, and $\boldsymbol{F}$ are trainable weight matrices, $\boldsymbol{w}$ is a trainable weight vector, and $\boldsymbol{b}$ is a trainable bias.

Finally, output mel-spectogram $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T_Y})$ and stop token $Z = (z_1, \ldots, z_{T_Y})$ are produced. For each time step t, $\boldsymbol{y}_t$ and z_t are calculated using the following equation:

$$\boldsymbol{y}_t = f_{FC}\left(\boldsymbol{s}_t\right), \tag{7}$$

$$z_t = f_{ST}\left(\boldsymbol{s}_t\right), \tag{8}$$

where f_{FC} is a fully connected network that processes the decoder state $\boldsymbol{s}_t$ by a linear projection to produce the predicted output and f_{ST} is a linear projection followed by sigmoid to predict when the production is stopped. A stacked convolutional post-net consumes $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T_Y})$ to obtain $Y' = (\boldsymbol{y}'_t, \ldots, \boldsymbol{y}'_{T_Y})$ by adding a residual prediction to improve the overall reconstruction as follows:

$$Y' = Y + postnet_{\theta_p}(Y). \tag{9}$$

2) MODEL T2-MLMS

T2-mlms model is an extension of T2 by adding d_L-dimensional language embedding $\boldsymbol{l}$ and d_S-dimensional speaker embedding $\boldsymbol{q}$. The embedding $\boldsymbol{l}$ and $\boldsymbol{q}$ are concatenated with the $T_X\_d_X$-dimensional text encoder hidden representation output $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx})$ before being consumed by decoder network, yielding $H' = (\boldsymbol{h}'_1, \ldots, \boldsymbol{h}'_{Tx})$ with $T_X\_(d_X + d_L + d_S)$-dimension, where for each $\boldsymbol{h}'_i$ is formulated as follows:

$$\boldsymbol{h}'_i = \boldsymbol{h}_i \oplus \boldsymbol{l} \oplus \boldsymbol{q}, \tag{10}$$

where $\oplus$ is a concatenation operator, $\boldsymbol{l}$ is the language embedding, and $\boldsymbol{q}$ is the speaker embedding. With this additional information, Equation (3) is changed into:

$$\boldsymbol{c}_t = \sum\nolimits_i^{T_X} \alpha_{t,i}\boldsymbol{h}'_i, \tag{11}$$

where $\boldsymbol{h}'_i$ is the concatenation of text, language, and speaker embedding. Likewise, Equation (5) is also changed into:

$$\boldsymbol{e}_{t,i} = \boldsymbol{w}^T tanh\left(\boldsymbol{W}\boldsymbol{s}_{t-1} + \boldsymbol{V}\boldsymbol{h}'_i + \boldsymbol{U}\boldsymbol{f}_{t,i} + \boldsymbol{b}\right). \tag{12}$$

3) MODEL T2-MLMS-GST

T2-mlms-gst model is an extension of T2-mlms by adding d_G-dimensional style embedding $\boldsymbol{g}$, d_P-dimensional pitch embedding, and $T_M\_T_X$-dimensional rhythm $\boldsymbol{R}$. Pitch P is extracted from the reference audio $M = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{T_M})$ with a length of T_M and $\boldsymbol{R}$ is the $T_M\_T_X$-dimensional alignment map between the reference audio M and the text. During the training process, the ground truth audio is used as the reference audio M and $\boldsymbol{R}$ is set with "none". Whereas during the model inference, $\boldsymbol{R}$ is extracted using T2-mlms-gst by performing teacher-forced forward pass from any desired reference audio M. The predicted mel-spectogram's length T_Y is equal to the reference mel-spectogram's length T_M because we apply force alignment.

In T2-mlms-gst model, the text hidden representation $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx})$ is concatenated with language embedding $\boldsymbol{l}$, speaker embedding $\boldsymbol{q}$, and style embedding $\boldsymbol{g}$ to produce $H'' = (\boldsymbol{h}''_1, \ldots, \boldsymbol{h}''_{Tx})$ with $T_X\_(d_X + d_L + d_S + d_G)$-dimension. Each $\boldsymbol{h}''_i$ is computed as follows:

$$\boldsymbol{h}''_i = \boldsymbol{h}_i \oplus \boldsymbol{l} \oplus \boldsymbol{q} \oplus \boldsymbol{g}, \tag{13}$$

Style embedding $\boldsymbol{g}$ is generated by the GST network from the reference audio and formulated as follows:

$$\boldsymbol{g} = GST_{\theta_G}(M), \tag{14}$$

where θ_G is the GST network parameters and $M = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{T_M})$ is the mel-spectogram of the reference audio. With this addition, in T2-mlms-gst model, Equation (3) is changed into:

$$\boldsymbol{c}_t = \sum\nolimits_i^{T_X} \alpha_{t,i}\boldsymbol{h}''_i, \tag{15}$$

where $\boldsymbol{h}''_i$ is the hidden representation of the text encoder concatenated with language, speaker, and style embedding. In here, the formulation between the model training and inference slightly differs. During training, the attention weight $\alpha_{t,i}$ is calculated using Equation (4) by changing Equation (5) into:

$$\boldsymbol{e}_{t,i} = \boldsymbol{w}^T tanh\left(\boldsymbol{W}\boldsymbol{s}_{t-1} + \boldsymbol{V}\boldsymbol{h}''_i + \boldsymbol{U}\boldsymbol{f}_{t,i} + \boldsymbol{b}\right). \tag{16}$$

Whereas during inference, the attention weight $\alpha_{t,i}$ is obtained from extracted alignment map $\boldsymbol{R}$, as follows:

$$\alpha_{t,i} = r_{t,i}, r_{t,i} \in \boldsymbol{R}. \tag{17}$$

Meanwhile, the pitch information $P = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{T_M})$ is extracted from the reference audio M using YIN algorithm that is processed through the pre-net-F0 decoder. Pitch $\boldsymbol{p}_t$ is concatenated with the previous spectogram output $\boldsymbol{y}_{t-1}$ that is processed through the pre-net decoder. This information is used by the decoder to find the decoder hidden state $\boldsymbol{s}_t$. It is calculated with a new equation, replacing Equation (2):

$$\boldsymbol{s}_t = decoder_{\theta_d}\left(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{p}_t, \boldsymbol{c}_t\right). \tag{18}$$

C. HIERARCHICAL TRANSFER LEARNING

Our models are trained using teacher-forcing procedure, the standard maximum-likelihood training, by feeding in the ground truth spectrogram frame instead of the predicted one to the decoder network. Thus, $\boldsymbol{y}_{t-1}$ in Equation (2) and (18) is replaced by ground truth $\boldsymbol{y}^{gt}_{t-1}$ during the training process. The model is optimized by minimizing the summed mean squared error (MSE) for the following objective function:

$$FLoss = \left(Y^{gt} - Y\right)^2 + \left(Y^{gt} - Y'\right)^2 + \left(Z^{gt} - Z\right)^2, \tag{19}$$

where Y^{gt} is the ground truth/target mel-spectogram, Y is the predicted mel-spectogram, Y' is the predicted mel-spectogram after post-net, Z^{gt} is the ground truth stop token sequence, and Z is the predicted stop token sequence.

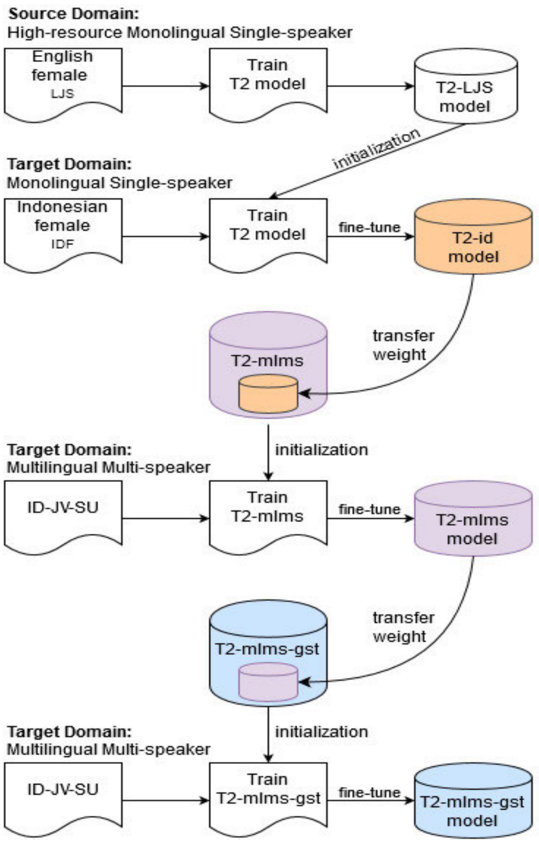


FIGURE 2. Hierarchical Transfer Learning for TTS. The first layer uses pre-trained T2 model on the English dataset, referred as T2-LJS. The parameters of T2-LJS are transferred to the model at the second layer and then it is fine-tuned on the Indonesian dataset resulting T2-id. The similar is done on the Javanese/Sundanese dataset to produce single-speaker monolingual T2-jv/T2-su. In the third layer, T2-id model parameters are transferred to initialize T2-mlms that is then fine-tuned on the joint multilingual dataset, ID-JV-SU. In the fourth layer, model T2-mlms-gst is partially initialized using pre-trained T2-mlms and fine-tuned using the same multilingual dataset.

There are four layers of training stages in our proposed hierarchical transfer learning architecture as illustrated in Figure 2. In the first layer, the monolingual single-speaker T2 model is trained on a high-resource language source domain. We use English source domain. In the second layer, the pre-trained monolingual single-speaker T2 model from the first layer is fine-tuned on a low-resource language target domain. We train our T2 model for Indonesian, Javanese, and Sundanese separately. In the third layer, the pre-trained model obtained on the second layer is transferred to initialize the multilingual multi-speaker T2-mlms model. Then, it is fine-tuned on a multilingual multi-speaker target domain. We use joint multilingual dataset of Indonesian, Javanese, and Sundanese languages. In the fourth layer, the pre-trained T2-mlms is transferred to partially initialize the multilingual multi-speaker with style transfer T2-mlms-gst model. Then, it is fine-tuned on the same joint dataset as target domain. Each model optimization in the hierarchical transfer learning scheme is shown in Algorithm 1 and Algorithm 2.

Algorithm 1 Hierarchical Transfer Learning for Multi-TTS

Input: Three different <text,audio> datasets: D(X_1, Y_1), D(X_2, Y_2), and D(X_3, Y_3), where D(X_2, Y_2) $\subset$ D(X_3, Y_3). D(X_1, Y_1) is a high-resource (source) language for single-speaker. D(X_2, Y_2) is a low-resource target language for single-speaker. D(X_3, Y_3) is a joint multilingual multi-speaker dataset of low-resource target languages.

1. Compute argmax $P(Y_1|X_1; \theta_1)$ where θ_1 is the T2 model parameters. Compute argmax $P(Y_2|X_2;\theta_2)$ where θ_2 is the T2 model parameters fine-tuned on θ_1.
2. Initialize T2-mlms parameters θ_{mlms_init} using standard initialization and transfer the weights from the corresponding T2 parameters θ_2 using Algorithm 2.
3. Compute argmax $P(Y_3|X_3, Q_3, L_3; \theta_{mlms})$ where Q_3 and L_3 are the speaker and language representations in the dataset D(X_3, Y_3), whereas θ_{mlms} are the T2-mlms model parameters fine-tuned on θ_{mlms_init}.
4. Initialize T2-mlms-gst parameters $\theta_{mlms\_gst\_init}$ and transfer the weights from the corresponding T2-mlms parameters θ_{mlms} using Algorithm 2.
5. Compute argmax $P(Y_3|X_3, Q_3, L_3, G_3, P_3; \theta_{mlms\_gst})$ where G_3 and P_3 are the speaking style and the pitch representations extracted from the reference audio (ground truth audio Y_3), whereas θ_{mlms_gst} is the model parameters fine-tuned on $\theta_{mlms\_gst\_init}$.

Output: θ_2, θ_{mlms}, and θ_{mlms_gst}

Algorithm 2 Transfer Parameter Weights for Multi-Models

```
Input: Model parameters target $\theta_{target}$ and model parameters source $\theta_{source}$, where structure($\theta_{source}$) $\subset$ structure($\theta_{target}$).
For each $\theta^s_w \subset \theta_{source}$ that corresponds to $\theta^t_w \subset \theta_{target}$, where $\theta^s_w$ and $\theta^t_w$ are trainable weight vectors, matrices, or 3D-tensors of a layer in our DNN models:
    If dimension($\theta^s_w$) $=$ dimension($\theta^t_w$):
        $\theta^t_w = \theta^s_w$.
    Else: #dimension($\theta^s_w$) $<$ dimension($\theta^t_w$)
        #update $\theta^t_w$ using element-wise update
        If $\theta^s_w$ is a vector, for each $w^s_a \in \theta^s_w$:
            $w^t_a = w^s_a$, $w^t_a \in \theta^t_w$
        Else-if $\theta^s_w$ is a matrix, for each $w^s_{a,b} \in \theta^s_w$:
            $w^t_{a,b} = w^s_{a,b}$, $w^t_{a,b} \in \theta^t_w$
        Else-if $\theta^s_w$ is a 3D-tensor, for each $w^s_{a,b,c} \in \theta^s_w$:
            $w^t_{a,b,c} = w^s_{a,b,c}$, $w^t_{a,b,c} \in \theta^t_w$
Output: model parameter target $\theta_{target}$.
```

The network-based deep transfer learning to the same model, such as T2 in the first and second layers, is quite simple. All network parameters of the pre-trained model on the source domain are transferred as initialized model in the next layer. Then, it is further fine-tuned on the target domain. If the transfer learning is from a simpler model to a more complex model, such as T2-mlms in the third and T2-mlms-gst in the fourth layers, an additional process is required to transfer the learned weights to a different model structure (see the second and fourth steps of Algorithm 1).

A different layer structure has a different trainable weight shape. In our models, the same layers with different structures are found in the attention and decoder. Hence, these layers have different dimension of weight matrices between prior model and success model. After the success model is created using the standard initialization, we transfer the prior model parameters to the corresponding success model parameters using Algorithm 2. The weight matrix (or vector or tensor) transfer learning, though only partially, is more effective than training the whole weight matrix from scratch. The partial weight matrix transfer allows us to fine-tune the higher dimension weight matrix of the success model by making use of the learned lower dimension weight matrix of the prior model.

IV. EXPERIMENTS

A. DATASET

Our work utilizes publicly available datasets: LJSpeech [50], an English speech corpus with a total duration of about 24 hours; TITML-IDN [51], an Indonesian (ID) speech corpus with an average of 43 minutes for each speaker; OpenSLR jv-ID [52], a Javanese (JV) speech corpus with an average of 10 minutes for each speaker; OpenSLR su-ID [52], a Sundanese (SU) speech corpus with an average of 7 minutes for each speaker.

TABLE 1. Single-speaker Monolingual Dataset in Indonesian, Javanese, and Sundanese.

Dataset	Corpus	Train	Dev	Test
IDF-01	TITML-IDN	38 min	5 min	5 min
JVF-06510	OpenSLR jv-ID	16 min	1 min	1 min
SUF-04190	OpenSLR su-ID	19 min	1 min	1 min

TABLE 2. Multi-speaker Multilingual Dataset in Indonesian, Javanese, and Sundanese.

Language	Corpus	Train	Dev	Test
ID	TITML-IDN	7.0 hr	42.0 min	4.5 min
JV	OpenSLR jv-ID	2.3 hr	7.8 min	2.5 min
SU	OpenSLR su-ID	1.7 hr	8.0 min	2.7 min
10ID-10JV-10SU		11.0 hr	57.8 min	9.7 min

T2 model for Indonesian, Javanese, and Sundanese uses a subset of corpus consisting of one female speaker for each language as shown in Table 1. Whereas for T2-mlms and T2mlms-gst, we use a joint multi-speaker multilingual dataset, referred as 10ID-10JV-10SU dataset as shown in Table 2. Data pre-processing was carried out to equalize the sample rate of audio to 16000 Hz and clean up text transcriptions.

B. MODEL IMPLEMENTATION

We implement our TTS model using PyTorch library [53]. For T2 model, we modify the open source code from NVIDIA Tacotron-2 [54] to support text processing for Indonesian, Javanese and Sundanese languages. For T2-mlms model, we add embedding networks to handle speaker and language identity. For T2-mlms-gst model, we add a reference encoder network for style embedding [23], GST network as in [24], and pitch and rhythm as in [25].

For each model, we use the same feature representations, both text and acoustic features. For text features, grapheme level is used to produce encoded text with a dimension of 512. We use 80 channels mel-spectrogram for the acoustic feature. We use a language embedding dimension of 8, a speaker embedding dimension of 128, a style embedding dimension of 256, and a 1-dimensional pitch embedding. More details about the spectral analysis and the model hyper-parameters can be seen in Table 3.

TABLE 3. The model Hyper-parameters of t2, t2-mlms, and t2-mlms-gst.

Spectral analysis	sample rate=16 KHz; N_FFT=1024; mel channel=80; window size=1024 (64 ms); hop size= 256 (16 ms); window type= Hann
Character embedding	512-D
Language embedding	8-D
Speaker embedding	128-D
GST embedding	256-D
Encoder	3 layer CNN: 512 filters with shape 5x1; 1 layer bi-LSTM: 512 units
Decoder pre-net	2 layer FCNN: FC-256-ReLU (Dropout=0.1)
Decoder pre-net F0	1 layer CNN: 1 filter with shape 1x1
Decoder RNN	2 layer LSTM: 1024 units
Attention RNN	1 layer LSTM: 1024 units
Decoder post-net	5 layer CNN: 512, 512, 512, 512, 80 filters with shape 5x1
GST	Reference encoder: 6 layer CNN: 32, 32, 64, 64, 128, 128 filters with shape 3x3 and stride 2x2; 1 layer GRU: 128 units Style token layer: Multi-head Attention= 8 heads; token=10

CNN: convolutional neural network, LSTM: long short-term memory, bi-LSTM: bidirectional LSTM, FCNN: fully connected neural network, GRU: gated recurrent unit. FFT: fast Fourier transform.

C. TRAINING SETUP

Each model is trained using two schemes: training from scratch and transfer learning. Each training scenario is carried out using a batch size of 32. We use 300K training steps except for T2 using transfer learning that uses 10K steps. The spectrogram prediction network training uses the standard maximum-likelihood (MLE) by feeding in the correct output instead of the prediction on the decoder side, referred as teacher-forcing. We use ADAM optimization [55] with default parameters, learning rates starting at 1e-3 and weight decay 1e-6. Models are trained using a single NVIDIA DGX-1 GPU.

Table 4 show the model names referred in this article along with the training setup information: the architecture, language, dataset, and pre-trained model for transfer learning.

TABLE 4. Model names, architectures, languages, datasets, and pre-trained models.

Model Name	Model Architecture	Language	Dataset	Pre-trained Model
T2-id-fs	T2	ID	IDF-01	-
T2-id-tl	T2	ID	IDF-01	T2-LJS
T2-jv-fs	T2	JV	JVF-06510	-
T2-jv-tl	T2	JV	JVF-06510	T2-id-tl
T2-su-fs	T2	SU	SUF-04190	-
T2-su-tl	T2	SU	SUF-04190	T2-id-tl
T2-mlms-fs	T2-mlms	ID-JV-SU	10ID-10JV-10SU	-
T2-mlms-tl	T2-mlms	ID-JV-SU	10ID-10JV-10SU	T2-id-tl
T2-mlms-gst-fs	T2-mlms-gst	ID-JV-SU	10ID-10JV-10SU	-
T2-mlms-gst-tl	T2-mlms-gst	ID-JV-SU	10ID-10JV-10SU	T2-mlms-tl

For T2 on Javanese and Sundanese, we use pre-trained T2 on IDF-01 instead of on LJS as explained in our prior work [45].

As for the vocoder, we use the pre-trained WaveGlow on LJS dataset [56] and fine-tune on IDF-01 dataset for single-speaker monolingual T2 model. Our experiments suggest that WaveGlow trained on the English LJS gives poor result when applied to Indonesian speech, so we need to fine-tune on Indonesian domain. Our experiments also conclude that WaveGlow trained on a female speaker can only produces good synthesize speech on the same gender speakers, but poor result on male speakers. Thus, for multi-speaker multilingual TTS, we further fine-tune WaveGlow on 10ID-10JV-10SU dataset.

D. MODEL EVALUATION

To evaluate our models, we use subjective assessments involving 9-20 respondents and objective assessments by measuring acoustic features. Two subjective evaluations are employed to measure the intelligibility of speech using semantically unpredictable sentences (SUS) [57] and to measure the quality of speech synthesis using a mean opinion score (MOS) [58] with scale of 1-5 with an increase of 1. Whereas the objective evaluations use four metrics as in [23]: mel-cepstral distortion ($\mathrm{MCD_K}$) [59], gross pitch error (GPE) [60], voicing decision error (VDE) [60], and F0 frame error (FFE) [61].

Before calculating the objective metrics, we apply padding according to the type of domain to equalize the length of signal frames because not all models produce the same signal length as of the reference audio. For $\mathrm{MCD_K}$ evaluation, we use 13 coefficients of mel-frequency cepstral coefficient (MFCC), producing the $\mathrm{MCD_{13}}$ metric. We extract pitch and voicing decisions using YIN algorithm [46] to calculate GPE, VDE, and FFE.

V. RESULTS AND ANALYSIS

This section presents the comparison of alignment learning using two training schemes: training from scratch and hierarchical transfer learning schemes for all models. It also presents the evaluation of the speech synthesis produced by the TTS models trained using the transfer learning scheme.

A. ALIGNMENT LEARNING

TTS is a seq-to-seq problem, when given text sequence it produces sound wave sequence. As a typical of seq-to-seq problems, it is important to learn the alignment between input sequence and output sequence. The TTS model that fails to do a reasonable alignment mapping is unable to synthesize intelligible speech that can be understood. We use location-sensitive attention scheme [47] to learn the mapping between input text and output mel-spectogram. This mapping is referred as an alignment map or attention map. TTS models that are able to produce intelligible speech can be indicated from the alignment that forms a diagonal map. In accordance with the nature of the TTS seq-to-seq problem using attention-based encoder decoder framework, the diagonal map shows that the alignment learning between the encoder steps and the decoder steps has been successful.

Our preliminary study suggests that training T2 model from scratch needs at least 10 hours of data to produce good quality of synthesized speech. Data under 3 hours was unable to produce clear synthesized speech and it is impossible for data below 1 hour to produce intelligible speech. However, for bigger scale models such as T2-mlms and T2-mlms-gst, 10 hours of training data is still insufficient. The standard training scheme fails to produce an accurate alignment map, hence the models are unable to produce intelligible speech. Different results are reported when we apply the proposed hierarchical transfer learning scheme. This scheme is able to learn fast and produce a reasonable alignment map using training data below 1 hour for T2 model (39, 16, and 18 minutes for Indonesian, Javanese, and Sundanese, respectively) and 11 hours data for T2-mlms and T2-mlms-gst. The learning process is shown in Figure 3 and the alignment maps are shown in Figure 4, Figure 5, and Figure 6 for T2, T2-mlms, and T2-mlms-gst, respectively.

These figures show the effectiveness of the transfer learning strategy applied on single-speaker monolingual TTS model T2 and multi-speaker multilingual with/without style transfer TTS models, T2-mlms-gst and T2-mlms. Using this learning scheme, all models can quickly learn the alignment. This is not the case for the standard training scheme, in which up to 300K iterations the model is still unable to produce reasonable map. Performing more iteration up to 500K does not enable T2-mlms-gst model to learn the proper mapping between text input and mel-spectogram output.

B. INTELLIGIBILITY AND NATURALNESS

The intelligibility and naturalness of the speech synthesized by TTS models are evaluated using SUS and MOS of a female speaker. The results are shown in Table 5.

Table 5 contains the results of MOS evaluations of T2 for Indonesian (ID), Javanese (JV), and Sundanese (SU), T2-mlms-tl, and T2-mlms-gst-tl. For SUS evaluations, we reports T2 and T2-mlms models only. For comparison,

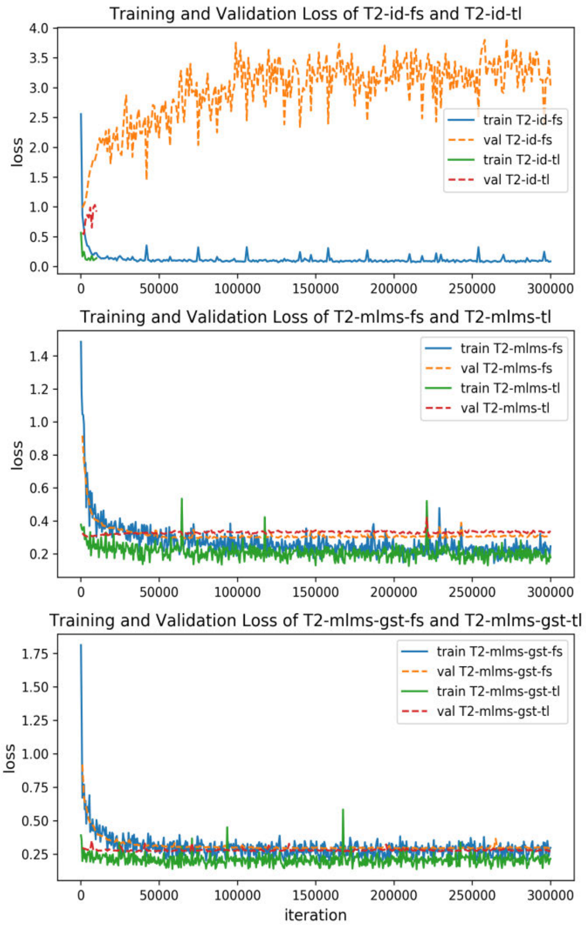


FIGURE 3. **The loss of Transfer Learning and Standard Training on T2, T2-mlms, and T2-mlms-gst. The loss charts are plotted starting at the 10-th iteration up to the 300K iteration, except for T2-id-tl that is up to 10K iterations. Learning process for all models converge starting at about 10K iterations. All graphs show that transfer learning converges faster and has better loss than standard training.**

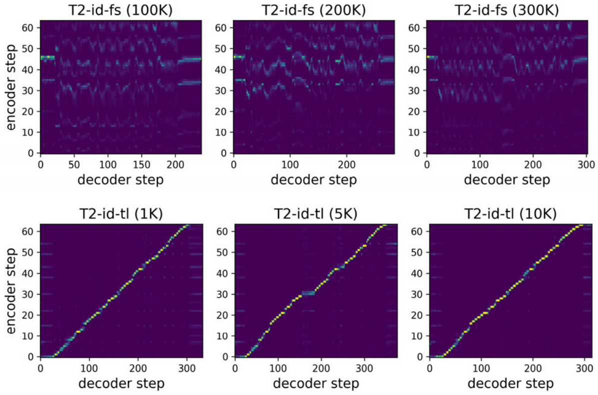


FIGURE 4. **Alignment Map of Standard Training and Transfer Learning of T2 model on IDF-01 dataset. Up to 300K iterations, the standard training (top) is still unable to learn the alignment properly. Whereas by using the transfer learning (bottom), the model can learn the alignment at the very early iterations (1K iterations). The training using transfer learning is done up to 10K iterations, because it is already convergent and produces good synthesized speech.**

the table also presents the MOS result of baseline Tacotron-2 [13] for English (ENG). For each language, the MOS of the real human voice is presented as the ground truth and a "diff" which contains the MOS difference between ground truth and synthesized speech produced by our models.

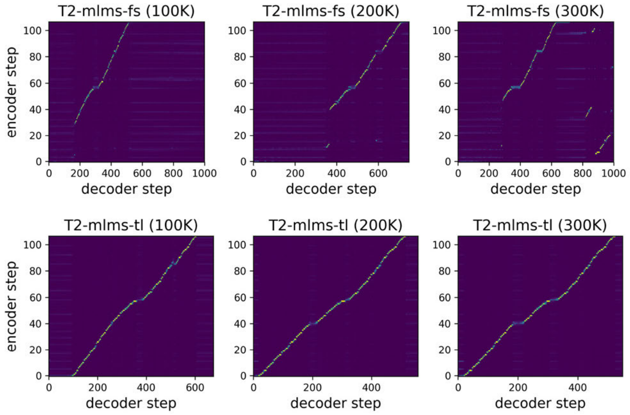


FIGURE 5. **Alignment Map of Standard Learning and Transfer Learning of T2-mlms on 10ID-10JV-10SU dataset. By using the standard training (top), the model fails to learn the alignment with iterations up to 300K. While by using the transfer learning (bottom) the model is able to learn the alignment successfully.**

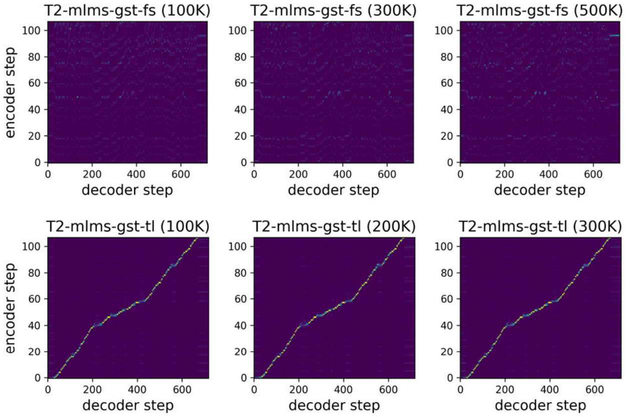


FIGURE 6. **Alignment Map of Standard Learning and Transfer Learning of T2-mlms-gst on 10ID-10JV-10SU dataset. By using the standard training scheme (top), the model fails to learn the alignment even with iterations up to 500K. While by using the transfer learning (bottom), it can produce a reasonable diagonal map.**

The MOS and SUS results demonstrate that only using training data less than 1 hour, our T2 model gives comparable MOS to the baseline Tacotron-2 trained on a large numbers of English dataset (24.6 hours). Likewise the more complex multi-speaker multilingual models, T2-mlms and T2-mlms-gst, can be trained using 11 hours of the joint multilingual dataset. Using the proposed learning scheme, our multilingual model provide even better "diff" value than of the English Tacotron-2 ("diff" = 0.056), specifically on Indonesian, and Sundanese with a "diff" of 0.017 and -0.08, respectively. The SUS evaluations show that our models are able to produce intelligible synthesized speech that can be understood. The SUS evaluation on Indonesian has the best performance with the word accuracy of 98.96%, whereas the SUS accuracy for Javanese and Sundanese are 98.52% and 97.53%, respectively.

TABLE 5. MOS and SUS-Wacc evaluation results.

Language	Model	MOS	Diff	SUS-Wacc
ENG	ground truth	4.582		
	Tacotron-2	4.526	0.056	-
ID	ground truth	4.362		
	T2-id-tl	4.271	0.090	98.26%
	T2-mlms-tl	**4.345**	**0.017**	**98.96%**
	T2-mlms-gst-tl	4.286	0.076	-
JV	ground truth	4.377		
	T2-jv-tl	4.077	0.300	95.02%
	T2-mlms-tl	**4.200**	**0.177**	**98.52%**
	T2-mlms-gst-tl	4.169	0.208	-
SU	ground truth	4.200		
	T2-su-tl	3.920	0.280	95.43%
	T2-mlms-tl	3.970	0.230	**97.53%**
	T2-mlms-gst-tl	**4.280**	**-0.080**	-

Overall, the SUS and MOS evaluations show that the performances of the model trained on the joint multilingual dataset are better than that of the model trained on the monolingual dataset. Using the joint multilingual dataset can significantly improve the naturalness and the intelligibility of the synthesized speech on each language. The model can generalize a language better by benefitting from other languages included in the joint multilingual dataset. Interestingly, Sundanese that has the least amount of data in the joint dataset (1.7 hours) has the most MOS improvement, an increment of 0.36 from T2-su-tl MOS to T2-mlms-gst-tl MOS on Sundanese, and gives better MOS than the ground truth. Whereas Javanese with 2.3 hours data in the joint dataset provides a MOS increment of 0.123 from T2-jv-tl MOS to T2-mlms-tl MOS on Javanese. As for Indonesian that shares 7 hours data, a MOS increment of 0.074 is obtained from T2-id-tl MOS to T2-mlms-tl MOS on Indonesian. In addition to the obvious benefit in using more data from other languages, we can see that the closer the phonetic similarity, the more benefit the language can gain. Sundanese that has only one additional phoneme of Indonesian's phonemes (while Javanese has three) obtains the most advantage from the phonetics similarity of Indonesian that shares the highest amount of data in the joint dataset, when it combines with style transferring.

C. PARALLEL STYLE TRANSFER

Parallel style transfer is the transfer of speaking style using the same sentence between the synthesized speech signal and reference signal. To evaluate the style transferring performance, we use GPE, VDE, FFE, and MCD_K metrics proposed by E2E-Prosody [23], each of which reflects the acoustic prosody correlation. For MCD_K evaluation we use MCD_{13} with $k = 13$ coefficients of MFCC.

Table 6 shows the GPE, VDE, FFE, and MCD evaluation results of speech synthesized by our models that are trained using transfer learning scheme: T2 for each language, T2-mlms, and T2-mlms-gst. The table also presents the evaluation results of E2E-Prosody [23] and Mellotron [25] for comparison. It also displays metrics per gender: F is for female, M is for male, and F/M is for both. The metrics are calculated by comparing the synthesized speech signal and the reference signal by speaker mentioned in "speaker" and "ref" columns, respectively. The synthesized speech are produced by the TTS model mentioned in "model" column.

1) PITCH TRACKING

Pitch tracking between synthesized speech and reference audio can be measured using FFE. FFE metric is a combination of two metrics: GPE that compares the pitch magnitude between the synthesized speech signals and the reference signal and VDE that compares the voicing decision (voiced/unvoiced). For prosody transfers, the lower the FFE the more successful style transfer is.

From Table 6 we can see that by applying style transfer, T2-mlms-gst model provides a better FFE measure compared to T2 and T2-mlms that do not apply it. The FFE results also demonstrate the effectiveness of our proposed hierarchical transfer learning to learn style transfer in T2-mlms-gst model. Using far less amount of training data (11 hours of joint multilingual dataset), our multilingual models give much better performance than monolingual E2E-Prosody that uses 147 hours and 296 hours training data for single-speaker and multi-speaker, respectively. In most cases, especially on female speakers, our T2-mlms-gst model is also better than Mellotron trained using 44 hours of LJS-Sally dataset and 41.7 hours of LibriTTS dataset. Moreover, our model is capable of transferring cross-lingual speakers that is not supported by both E2E-prosody and Mellotron. Our model allows speakers of one language to speak fluently in other languages. However, we can see there is gender bias in FFE results: FFE on male speakers are slightly worse than FFE on female speakers.

Figure 7 shows the comparison of pitch tracking between synthesized speech signal and reference signal for the same sentence in each language. We can see that the model applying prosody transfer, T2-mlms-gst-tl, is able to imitate the reference pitch contours well. T2-mlms-tl that does not apply prosody transfer produces different contours.

2) MEL-SPECTOGRAM

MCD_{13} is a metric for measuring distortion between synthesized signal and reference signal using 13 coefficients of MFCC. Lower score of MCD_{13} has better performance. From Tables 6, the MCD_{13} scores of our models for Indonesian and Sundanese are better than E2E-Prosody's. Different from FFE gender bias, the MCD_{13} scores differ among speakers regardless their gender. MFCC is computed using discrete cosine transform (DCT) operation on mel-spectogram.

TABLE 6. GPE, VDE, FFE, and MCD for multi-speaker.

ref	speaker	model	GPE			VDE			FFE			MCD_{13}		
			F	M	F/M	F	M	F/M	F	M	F/M	F	M	F/M
SS	Single-S	E2E-Prosody	-	-	-	-	-	-	28.10%	-	-	7.92	-	-
SS	Multi-S	E2E-Prosody	-	-	-	-	-	-	-	-	27.50%	-	-	6.99
DS	Multi-S	E2E-Prosody	-	-	-	-	-	-	-	-	37.10%	-	-	9.51
SS	LJS-Sally	Mellotron	0.26%	-	-	9.19%	-	-	9.28%	-	-	-	-	-
	LibriTTS	Mellotron	-	-	0.42%	-	-	8.68%	-	-	8.77%	-	-	-
SS	IDF	T2-id-tl	29.58%	-	-	40.43%	-	-	51.56%	-	-	12.57	-	-
SS	JVF	T2-jv-tl	14.36%	-	-	29.86%	-	-	38.61%	-	-	**2.69**	-	-
SS	SUF	T2-su-tl	8.99%	-	-	22.20%	-	-	28.59%	-	-	**3.85**	-	-
ID-S	Multi-ID	T2-mlms-tl	20.99%	31.33%	26.16%	44.15%	43.47%	43.81%	50.71%	48.80%	49.75%	**4.40**	**0.06**	**2.23**
		T2-mlms-gst-tl	0.81%	3.80%	2.31%	**6.51%**	9.87%	**8.19%**	**6.93%**	11.24%	**9.08%**	**4.83**	**0.01**	**2.42**
	Multi-JV	T2-mlms-tl	39.58%	30.05%	34.81%	44.66%	45.71%	45.18%	57.48%	51.57%	54.53%	**4.85**	**0.11**	**2.48**
		T2-mlms-gst-tl	1.35%	3.05%	2.20%	**7.51%**	11.37%	9.44%	**8.20%**	12.39%	10.30%	**5.23**	**0.01**	**2.62**
	Multi-SU	T2-mlms-tl	29.81%	36.51%	33.16%	42.13%	45.07%	43.60%	50.82%	51.94%	51.38%	**4.81**	**0.41**	**2.61**
		T2-mlms-gst-tl	0.91%	2.77%	**1.84%**	**7.51%**	11.05%	9.28%	**7.99%**	12.00%	9.99%	**4.72**	**0.01**	**2.36**
JV-S	Multi-ID	T2-mlms-tl	43.54%	36.65%	40.09%	33.86%	42.02%	37.94%	56.89%	55.28%	56.09%	10.02	14.98	12.50
		T2-mlms-gst-tl	1.39%	3.17%	2.28%	**8.59%**	12.27%	10.43%	9.54%	13.76%	11.65%	**8.52**	17.26	12.89
	Multi-JV	T2-mlms-tl	12.91%	23.30%	18.10%	31.53%	37.37%	34.45%	38.78%	45.70%	42.24%	**5.65**	8.78	**7.21**
		T2-mlms-gst-tl	1.48%	2.14%	**1.81%**	**7.54%**	10.56%	**9.05%**	**8.61%**	11.65%	**10.13%**	**6.72**	16.35	11.54
	Multi-SU	T2-mlms-tl	43.52%	39.66%	41.59%	37.75%	41.42%	39.58%	59.82%	55.15%	57.48%	**8.97**	15.06	12.02
		T2-mlms-gst-tl	1.64%	2.30%	1.97%	10.08%	13.55%	11.82%	11.14%	14.64%	12.89%	**8.45**	16.15	12.30
SU-S	Multi-ID	T2-mlms-tl	32.10%	25.22%	28.66%	31.85%	36.64%	34.25%	50.71%	47.40%	49.05%	**4.28**	**4.86**	**4.57**
		T2-mlms-gst-tl	0.54%	**0.11%**	**0.32%**	**6.98%**	9.69%	**8.34%**	**7.41%**	9.77%	**8.59%**	**3.48**	**4.89**	**4.19**
	Multi-JV	T2-mlms-tl	36.29%	34.35%	35.32%	28.59%	39.37%	33.98%	50.55%	54.05%	52.30%	**4.00**	**5.22**	**4.61**
		T2-mlms-gst-tl	0.51%	**0.27%**	**0.39%**	**6.60%**	10.01%	**8.30%**	**6.99%**	10.15%	**8.57%**	**3.48**	**4.87**	**4.17**
	Multi-SU	T2-mlms-tl	11.32%	8.48%	9.90%	25.37%	31.04%	28.21%	32.83%	35.11%	33.97%	**1.98**	**3.06**	**2.52**
		T2-mlms-gst-tl	0.53%	**0.18%**	**0.36%**	**6.52%**	9.80%	**8.16%**	**6.93%**	9.93%	**8.43%**	**3.02**	**5.04**	**4.03**

Ref is for reference, SS: same speaker, DS: different speaker, ID-S: Indonesian speakers, JV-S: Javanese speakers, SU-S: Sundanese speakers. Speaker denotes the speakers of the synthesized speech, IDF: an Indonesian female, JVF: a Javanese female, SUF: a Sundanese female, Multi-ID: multi Indonesian speakers, Multi-JV: multi Javanese speakers, Multi-SU: multi Sundanese speakers. Bold and blue marked fonts indicate the value is higher than the baseline models, GPE/VDE/FFE value that is higher than Mellotron's and MCD value that is higher than E2E-Prosody's. Red fonts indicate the best F/M performance in the group.

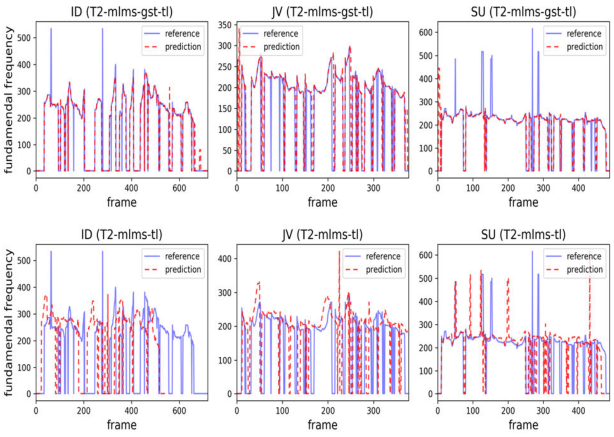


FIGURE 7. Pitch Contour Comparison between Reference and Synthesized speech by T2-mlms-gst-tl (top) and T2-mlms-tl (bottom) for Indonesian (left), Javanese (middle), and Sundanese (right). The sentences used are "hanya dalam waktu kurang dari lima menit tamara bleszinsky muncul kembali di panggung dengan busana berbeda" for Indonesian (ID), "gedhung ingkang regine gangsal triliun niku kados napa rupanipun" for Javanese (JV), and "dina bulan silih mulud taun dua rebu lima belas seueur umat islam nu ngalakonan ibadah umroh" for Sundanese (SU).

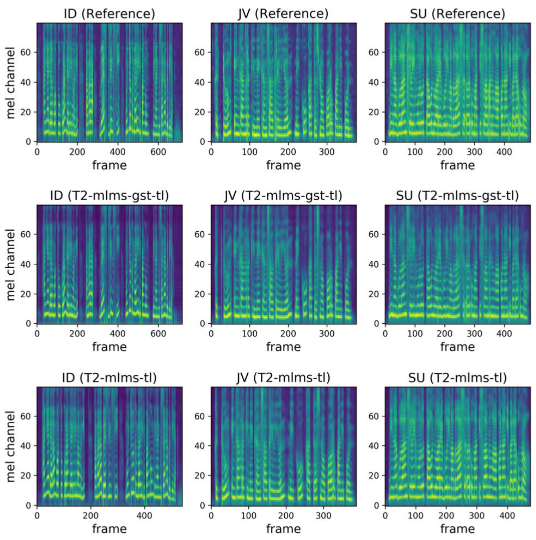
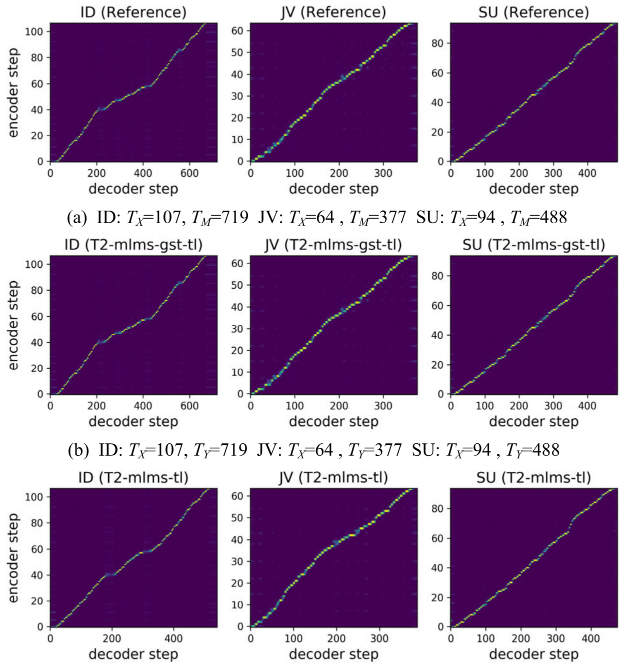


FIGURE 8. Mel-spectogram of the Reference Audio (top) and Synthesized Speech by t2-mlms-gst (middle - vertical) and t2-mlms (bottom) for Indonesian (left), Javanese (middle - horizontal), and Sundanese (right). T_X is the length of the embedding input text, T_Y is the length of the predicted mel-spectogram by the models, and T_M is the length of the reference mel-spectogram. The sentences used are the same as in Figure 7.

The mel-spectogram comparison between reference audio and the synthesized audio of the same texts is illustrated in Figure 8.

3) RHYTHM

Figure 9 shows the alignment map between text sentences as used in Figure 8 and their corresponding speech signals



(a) ID: T_X=107, T_M=719 JV: T_X=64 , T_M=377 SU: T_X=94 , T_M=488

(b) ID: T_X=107, T_Y=719 JV: T_X=64 , T_Y=377 SU: T_X=94 , T_Y=488

(c) ID: T_X=107, T_Y=541 JV: T_X=64 , T_Y=388 SU: T_X=94 , T_Y=476

FIGURE 9. Alignment Map of the Reference Audio (top) and Synthesized Speech by t2-mlms-gst (middle - vertical) and t2-mlms (bottom) for Indonesian (left), Javanese (middle - horizontal), and Sundanese (right).

in Indonesian, Javanese and Sundanese. The same texts, represented as encoder steps, are mapped to the reference signals (a), the predicted speech signals by T2-mlms-gst (b), the predicted speech signals by T2-mlms (c). From this figure, we can also see that the synthesis by T2-mlms model has a different number of decoder steps from the reference signal's, while using forced-alignment by feeding the rhythm to T2-mlms-gst model can produce the same decoder steps as the reference's. The higher the number of decoder steps the slower the rhythm, and vice versa, the fewer the decoder steps the faster the rhythm of the speech.

VI. CONCLUSION

Our work develops Tacotron-2-based multi-speaker multilingual TTS with/without style transfer by adding several new components: speaker embedding, language embedding, style embedding, pitch embedding, and rhythm. To train the models, we propose hierarchical transfer learning, a network-based transfer learning, that benefits from previous learning on a high-resource (source) language. Pre-trained model parameters are transferred to the same model that is fine-tuned on a low-resource (target) language and to a more complex model that is fine-tuned on a joint multilingual dataset with phonetic similarity.

From the experiment results, we demonstrate that the hierarchical transfer learning scheme is an effective choice to be applied in low-resource target languages. The alignment learning, that is crucial in attention-based encoder-decoder TTS model, is successfully transferred from source to target domain by fine-tuning the pre-trained source model on a small amount of target data. Moreover, the model can benefit from using a joint multilingual dataset for better generalization. The TTS multilingual models are able to generate intelligible human-like synthesized speech. In addition, our multi-speaker multilingual with style transfer TTS is able to adequately transfer the speaking style of one speaker to another speaker of the same language or different ones.

Despite having high performance on a joint multilingual dataset with phonetic similarity, it is challenging to study the transfer learning strategy on a low-resource domain using a multilingual dataset with high differences in linguistic aspects such as phonetics, phonology, and grapheme symbol diversity.

ACKNOWLEDGMENT

The authors thank Tokopedia-UI AI Center team and Lab 1231 Fasilkom UI team for their helpful discussion and feedback.

REFERENCES

[1] W. Mattheyses and W. Verhelst, "Audiovisual speech synthesis: An overview of the state-of-the-art," *Speech Commun.*, vol. 66, pp. 182–217, Feb. 2015.

[2] D. Jurasfky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.

[3] P. Taylor, *Text-to-Speech Synthesis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[4] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Appl. Sci.*, vol. 9, no. 19, p. 4050, Sep. 2019.

[5] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proc. IEEE*, vol. 101, no. 5, pp. 1234–1252, May 2013.

[6] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 1, pp. 66–83, Jan. 2009.

[7] T. Koriyama and T. Kobayashi, "Statistical parametric speech synthesis using deep Gaussian processes," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 5, pp. 948–959, May 2019.

[8] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. 38th IEEE Int. Conf. Acoust. Speech, Signal Process.*, May 2013, pp. 7962–7966.

[9] S. Achanta and S. V. Gangashetty, "Deep Elman recurrent neural networks for statistical parametric speech synthesis," *Speech Commun.*, vol. 93, pp. 31–42, Oct. 2017.

[10] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, pp. 1–67, 2019.

[11] A. Khamparia and K. M. Singh, "A systematic review on deep learning architectures and applications," *Expert Syst.*, vol. 36, no. 3, pp. 1–22, 2019.

[12] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[13] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4779–4783.

[14] 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 *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6905–6909.

[15] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End speech synthesis," in *Proc. Interspeech*, Aug. 2017, pp. 4006–4010.

[16] W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "Wavenet based low rate speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 676–680.

[17] A. Van Den Oord *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 9, 2018, pp. 6270–6278.

[18] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2017, pp. 264–273.

[19] S. O. Arik *et al.*, "Deep voice 2: Multi-speaker neural text-to-speech," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 2963–2971.

[20] W. Ping *et al.*, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[21] Y. Zheng, J. Tao, Z. Wen, and J. Yi, "Forward–backward decoding sequence for regularizing end-to-end TTS," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2067–2079, Dec. 2019.

[22] Y. Liu and J. Zheng, "ES-tacotron2: Multi-task tacotron 2 with pre-trained estimated network for reducing the over-smoothness problem," *Information*, vol. 10, no. 4, p. 131, Apr. 2019.

[23] R. J. 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 *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 11, 2018, pp. 7471–7480.

[24] 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," 2018, *arXiv:1803.09017*. [Online]. Available: http://arxiv.org/abs/1803.09017

[25] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6189–6193.

[26] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. 27th Int. Conf. Artif. Neural Netw.*, 2018, pp. 2672–2680.

[27] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. J. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6940–6944.

[28] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. ASRU*, Dec. 2017, pp. 301–308.

[29] D.-R. Liu, C.-Y. Yang, S.-L. Wu, and H.-Y. Lee, "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Dec. 2018, pp. 640–647.

[30] S. Novitasari, A. Tjandra, S. Sakti, and S. Nakamura, "Cross-lingual machine speech chain for Javanese, Sundanese, Balinese, and Bataks speech recognition and synthesis," in *Proc. 1st Joint Workshop Spoken Lang. Technol. Under-Resour. Lang. (SLTU)*, May 2020, pp. 131–138.

[31] Y.-J. Chen, T. Tu, C.-C. Yeh, and H.-Y. Lee, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning," in *Proc. Interspeech*, Sep. 2019, pp. 2075–2079.

[32] S. Jialin Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[33] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, 2016, Art. no. 9.

[34] Y. Wu, X. Qin, Y. Pan, and C. Yuan, "Convolution neural network based transfer learning for classification of flowers," in *Proc. IEEE 3rd Int. Conf. Signal Image Process. (ICSIP)*, Jul. 2018, pp. 562–566.

[35] I. Kandel and M. Castelli, "Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review," *Appl. Sci.*, vol. 10, no. 6, p. 2021, Mar. 2020.

[36] G. Luo, Y. Yang, Y. Yuan, Z. Chen, and A. Ainiwaer, "Hierarchical transfer learning architecture for low-resource neural machine translation," *IEEE Access*, vol. 7, pp. 154157–154166, 2019.

[37] Y. Chen, Y. Liu, Y. Cheng, and V. O. K. Li, "A teacher-student framework for zero-resource neural machine translation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2017, pp. 1925–1935.

[38] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of language-independent End-to-end ASR with language model fusion," in *Proc. ICASSP*, May 2019, pp. 6096–6100.

[39] J. Yi, J. Tao, Z. Wen, and Y. Bai, "Language-adversarial transfer learning for low-resource speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 3, pp. 621–630, Mar. 2019.

[40] C. Yu, Y. Chen, Y. Li, M. Kang, S. Xu, and X. Liu, "Cross-language end-to-end speech recognition research based on transfer learning for the low-resource Tujia language," *Symmetry*, vol. 11, no. 2, p. 179, Feb. 2019.

[41] K. Feng and T. Chaspari, "Low-resource language identification from speech using transfer learning," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Oct. 2019, pp. 1–6.

[42] X. Wei, H. Lin, Y. Yu, and L. Yang, "Low-resource cross-domain product review sentiment classification based on a CNN with an auxiliary large-scale corpus," *Algorithms*, vol. 10, no. 3, p. 81, Jul. 2017.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[44] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Bright. U.K., May 2019, pp. 3617–3621.

[45] K. Azizah and M. Adriani, "Hierarchical transfer learning for text-to-speech in Indonesian, Javanese, and Sundanese languages," in *Proc. 12th Int. Conf. Adv. Comput. Sci. Inf. Syst.*, 2020.

[46] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Amer.*, vol. 111, no. 4, pp. 1917–1930, Apr. 2002.

[47] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2015, pp. 577–585, 2015.

[48] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015, pp. 1–15.

[49] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2018, pp. 10215–10224.

[50] K. Ito. (2017). *The LJ Speech Dataset 2017*. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[51] D. Lestari, K. Shinoda, and S. Furui, "A large vocabulary continuous speech recognition system for Indoneia Language," in *Proc. 15th Indonesian Sci. Conf. Japan*, 2006, pp. 17–22.

[52] K. Sodimana, P. De Silva, S. Sarin, O. Kjartansson, M. Jansche, K. Pipatsrisawat, and L. Ha, "A step-by-step process for building TTS voices using open source data and frameworks for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese," in *Proc. 6th Int. Workshop Spoken Lang. Technol. Under-Resour. Lang.*, Aug. 2018, pp. 66–70.

[53] A. Paszke, S. Gross, S. Chintala, and G. Chanan. *PyTorch*. Accessed: Feb. 10, 2020. [Online]. Available: https://pytorch.org/PyTorch

[54] NVIDIA. (2018). *Tacotron 2 Without WaveNet*. [Online]. Available: https://github.com/ NVIDIA/tacotron2

[55] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. ICLR (Poster)*, 2015.

[56] NVIDIA. (2018). *WaveGlow*. [Online]. Available: https://github.com/NVIDIA/WaveGlow

[57] M. Grice and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences," *Speech Commun.*, vol. 6393, pp. 381–392, Jun. 1996.

[58] Y.-Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Jan. 2003, pp. 577–582.

[59] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. IEEE Pacific Rim Conf. Commun. Comput. Signal Process.*, vol. 1, May 1993, pp. 125–128.

[60] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo, "A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments," *Speech Commun.*, vol. 50, no. 3, pp. 203–214, Mar. 2008.

[61] W. Chu and A. Alwan, "Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 3969–3972.

KURNIAWATI AZIZAH (Member, IEEE) received the B.S. degree in informatics engineering from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 1997, and the M.Phil. degree in computer speech, text, and Internet technology (CSTIT) from the University of Cambridge, Cambridge, U.K., in 2006. She is currently pursuing the Ph.D. degree in computer science with Universitas Indonesia, Jakarta, Indonesia.

She was an IT Consultant with MINCOM Indoservices, Jakarta, from 1998 to 2000, and Switchlab, London, U.K., from 2000 to 2017. Since 2008, she has been a Lecturer with the Faculty of Computer Science, Universitas Indonesia. Her research interests include deep learning, natural language processing (NLP), speech processing, and computer vision.

MIRNA ADRIANI (Member, IEEE) received the B.S. degree from California State University, Pomona, CA, USA, and the Ph.D. degree in computer science from Glasgow University, Glasgow, U.K.

From 1998 to 2000, she was a Visiting Researcher with the Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, USA. From 2006 to 2020, she was the Head of the Information Retrieval Laboratory, Faculty of Computer Science, Universitas Indonesia. From 2014 to 2020, she was the Dean of the Faculty of Computer Science, Universitas Indonesia. Her research interests include speech processing and information retrieval (IR), cross-language IR, summarization, question answering, microblog analysis, and music IR.

WISNU JATMIKO (Senior Member, IEEE) received the B.S. degree in electrical engineering and the M.Sc. degree in computer science from Universitas Indonesia, in 1997 and 2000, respectively, and the Dr.Eng. degree from Nagoya University, Japan, in 2007.

He currently works as a Full Professor with the Faculty of Computer Science, Universitas Indonesia. His research interests include autonomous robot, optimization, and real-time traffic monitoring systems. He is also the Chair of the IEEE Indonesia Section.