Master's Thesis

Transfer Learning for Sequence-to-Sequence Voice Conversion

251906183 Wen-Chin Huang

Department of Intelligent Systems

Graduate School of Informatics

Nagoya University

February 2021

Contents

Abstract

Voice conversion (VC) is a technique that can transform the characteristics of the source speech into that of the target speech such that the linguistic contents are preserved. As conventional VC models perform conversion in a frame-wise manner, i.e. the converted speech always has the same length as that of the source speech, VC based on sequence-to-sequence (seq2seq) modeling has become attractive in recent years, owing to their ability to convert suprasegmental characteristics such as prosody and speaker rate. Nonetheless, one major problem is that the practical amount of data in VC is too limited for seq2seq models. Such data-hungry property causes low intelligibility in the converted speech, making seq2seq VC far from practical. In this thesis, we study how to apply the technique of transfer learning, which is a common yet effective concept in machine learning that alleviates data deficiency.

In the first part of this thesis, we present a naive approach that concatenates an automatic speech recognition (ASR) model and a text-to-speech (TTS) model. The input speech is first transcribed with the ASR model, and the recognition result is then used to generate the voice of the target with the TTS model. Thanks to the massive amount of data and the mature development in the two respective fields, VC can benefit from transferring knowledge of readily trained ASR and TTS models, thus improving data efficiency. This method was employed as the seq2seq baseline system for the voice conversion challenge (VCC) 2020, and we utilized ESPnet, an open-source end-to-end speech processing toolkit, and the many well-configured pretrained models provided by the community. We demonstrate that such a method can successfully perform conversion using only 5 min of training data, and the official evaluation results showed that the system came out top among the participating teams in terms of conversion similarity, serving as a strong baseline system.

In the second part, instead of separately training two models, we proposed to optimize one unified model by a novel pretraining technique that, similarly, transfers knowledge from ASR and TTS. We argued that the essential ability of a VC model, i.e, the generation and utilization of fine representations, can be facilitated by a two-step pretraining-finetuning scheme using either ASR or TTS. The pretraining process provides a prior for fast, sample-efficient VC model learning, thus reducing the data size requirement and training time. We demonstrate that the VC model initialized with pretrained model parameters can generate high-quality, highly intelligible speech even with limited training data.

1 Introduction

1.1 General background

Voice conversion (VC) aims to convert the speech from a source to that of a target without changing the linguistic content [1]. Speaker voice conversion [2] is a typical type of VC and refers to the process of converting speech from a source speaker to a target speaker. In addition, a wide variety of applications could be solved by applying VC, such as accent conversion [3], personalized speech synthesis [4,5], and speaking-aid device support [6–8].

As an ultimate goal, unconstrained speech communication is one of the most important application of VC. The physical condition of the human body often limits the production of speech [9] as shown in Figure 1.1. For instance, damaged speech organs cause severe vocal disorders. Or, the deficient control of the organs can end up with an accented voice while the intention is to speak a foreign language natively. What if we can recover disabled functions, or even augment our body to enhance communication abilities? By building VC systems such as speaking aid devices to convert electrolaryngeal (EL, restored using an electronic device) speech to the original voices of patients with vocal cord damage [6] or converting accented speech of foreigners into native speech [10], speech communication can be made beyond physical constraints.

A typical framework of VC adopts an *analysis—conversion —synthesis* paradigm [11], as decipted in Figure 1.2. First, a high-quality vocoder such as WORLD [12]

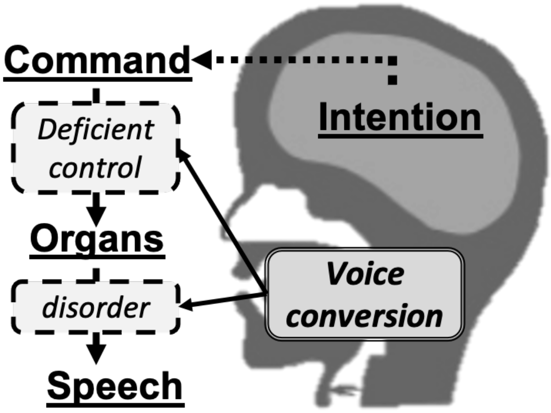


Figure 1.1: *Illustration of the limits of speech communication, and how voice conversion can break the barrier.*

or STRAIGHT [13] is utilized to extract different acoustic features, such as spectral features and fundamental frequency (F0). These features are converted separately, and a waveform synthesizer finally generates the converted waveform using the converted features.

Numerous VC approaches have been proposed. The Gaussian mixture model (GMM)-based method [11, 14] has been a popular statistical approach that estimates the joint density of the source-target feature vectors, which requires a training procedure and has a well-known disadvantage that the converted outputs generally suffer from an over-smoothing issue. Frequency warping methods, such as vocal tract length normalization [15], weighted frequency warping [16] and dynamic frequency warping [17], are able to keep spectral details while providing inferior speaker identity conversion quality to that of statistical approaches. Exemplar-based methods [18–22] require much less training data and are capable of modeling the high-dimensional spectra. In recent years, deep neural networks (DNNs) have established supremacy in a wide range of

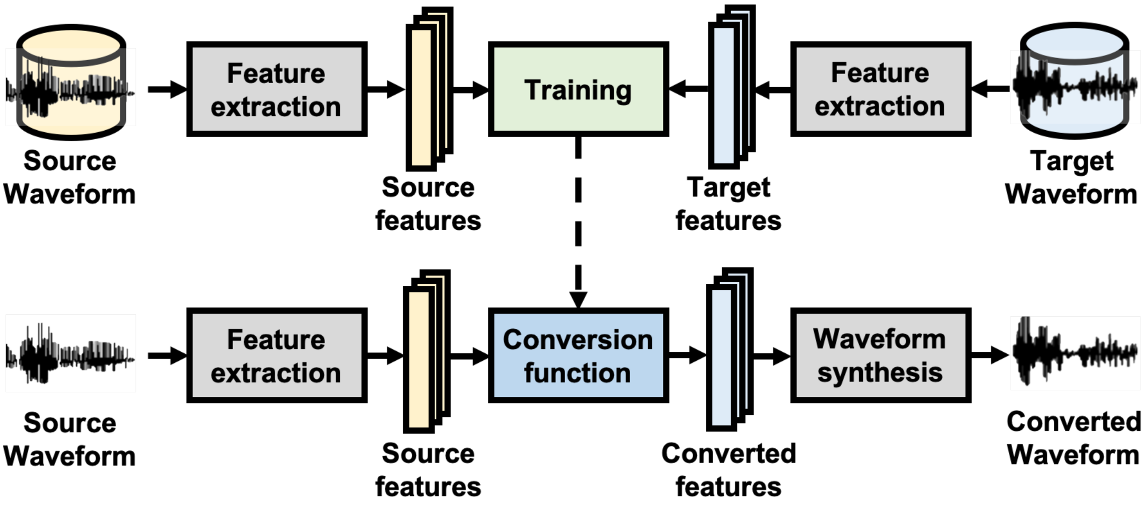


Figure 1.2: *A general VC framework realizing the analysis—conversion —synthesis paradigm.*

research fields, including VC [23–26]. DNNs have been utilized for not only spectral mapping but also neural vocoding [27–29]. It has been shown that employing neural vocoders as the waveform generation module can greatly improve the performance of VC systems [30–35].

However, most of the approaches described above suffer from two limitations. First, it was long believed that the spectral property plays an important role in characterizing speaker individuality. As a result, as shown in Figure 1.3, most efforts were dedicated in spectral conversion, while only a simple linear transformation was applied to F0, limiting the modeling of various speaker characteristics. Second, conversion was usually performed frame-by-frame, i.e, the converted speech and the source speech share the same length as well as temporal structure. This restricts the modeling of the speaking rate and style such as short pause pattern. To summarize, the conversion of prosody, including F0 and duration, is overly simplified in the literature.

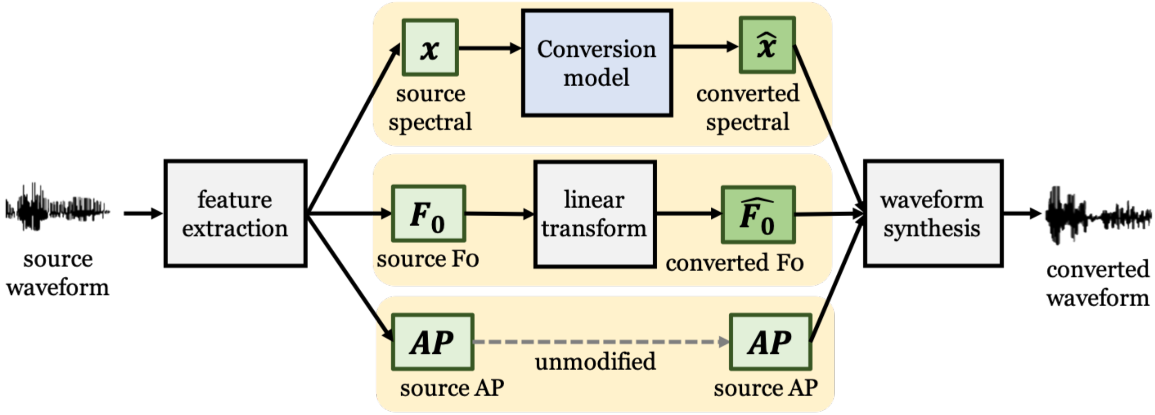


Figure 1.3: *Illustration of a VC system performing conversion w.r.t. different feature streams.*

This is where sequence-to-sequence (seq2seq) models [36] can play a role. Modern seq2seq models, often equipped with an attention mechanism [37,38] to implicitly learn the alignment between the source and output sequences, can generate outputs of various lengths and capture long-term dependencies. This ability makes the seq2seq model a natural choice to convert prosody in VC. It has been shown that seq2seq VC models can outperform conventional frame-wise VC systems, especially in terms of conversion similarity [39–41]. This is owing to the fact that the suprasegmental characteristics of F0 and duration patterns well handled in seq2seq VC models are closely correlated with the speaker identity.

Despite the promising results, seq2seq VC models suffer from a data-hungry property. In the literature, seq2seq models usually require a large amount of training data to generalize well, with a requirement of more than 1000 utterances (approximately 1 hour) of data. However, it is impractical to collect such a large parallel VC corpus. For example, in the past voice conversion chellenges (VCCs) [42–44], at most 160 utterances (approximately 10 min of speech data) were given, which is much less than the 1000 utterances required by seq2seq models. As a result, as pointed out in [45], the attention learning fails as the amount of training data decreases, causing mispronunciations and other linguistic-inconsistency problems such as inserted, repeated and skipped phonemes in the converted speech. It is therefore essential to develop sample efficient seq2seq VC systems.

One popular means of dealing with the problem of limited training data is transfer leaning, where knowledge from massive, out-of-domain data is utilized to aid learning in the target domain. Thanks to the big data era, transfer learning has been made easy since it is easy to collect a huge amount of data. In this thesis, we propose two novel appraoches that solves the data deficiency problem by transfering knowledge from two speech processing tasks, namely automatic speech recognition (ASR) and text-to-speech (TTS).

1.2 Related work

1.2.1 Transfer learning from automatic speech recognition and text-to-speech to voice conversion

In speech processing, ASR and TTS are two of the most active research fields, and the research community has dedicated an enumerous effort to open-sourcing public available datasets. It was shown, in [46], that a sufficient amount of efforts has been dedicated to transferring knowledge from ASR [47–51] and TTS [52–54].to improving various aspects of VC, regardless of using a seq2seq model or not. However, few of the abovementioned studies showed how the proposed methods can solve the data deficiency problem in seq2seq VC, which is the main goal of this thesis.

1.2.2 Data deficiency in sequence-to-sequnece voice conversion

In VC, it is common to limit the size of training data to around 5 or 10 minutes [42,43], wherein existing seq2seq VC literature, approximately 1 hour of data is usually used. Even with such amount of data, it is still required to resort to certain techniques to successfully train seq2seq VC models. These techniques can be categorized into the followings:

Extra module. Many have utilized an external ASR module pretrained on a large dataset during training and runtime. For example, the phonetic posteriorgram (PPG) extracted from ASR is a commonly used linguistic clue in VC [55], and can be used as the only input [56] or as an additional clue [40,45] in seq2seq VC. On the other hand, Parrotron [57] used an external TTS system to generate artificial data from a large hand-transcribed corpus for training an any-to-one (normalization) VC model. The disadvantage of using external modules is that the performance depends on the extra module. The accuracy of the PPG and the quality of the TTS system can bound the performance of the final VC system.

Text label of training data. Text labels provide strong supervision to ensure linguistic consistency. Methods utilizing such labels include multitask learning meaningful hidden representation [45,57,58], data augmentation [45] or representations disentanglement [58]. Yet, labeling errors and failed force alignments might cause potential performance degradation.

Regularization. As multitask learning and feature disentanglement can be seen as regularizations, some have also proposed to impose constraints on the model without any external resources. [39,59] proposed the context preservation loss and the guided attention loss, and [56] proposed to use local attention to stabilize training. Nonetheless, such regularization often requires rigorous weight tuning.

1.3 Thesis scope

In this thesis, we propose two methods for transferring knowledge from ASR and TTS to tackle the data deficiency problem in seq2seq VC. To better understand the motivation of these two methods, we first provide a unified perspective of voice conversion w.r.t. the information in speech. Then, we elaborate on how the two approaches proposed in this thesis connect to the information perspective. Finally, in Table 1.1 we compare the two methods w.r.t. the three techniques described in Section 1.2.2 for tackling the data deficiency problem in seq2seq VC.

1.3.1 Information perspective of voice conversion

Roughly speaking, speech consists of the linguistic contents and the speaker identity. The goal of VC is to remove the source speaker information from the source speech, and then inject the identity of the target speaker. As depicted in Figure 1.4, an ideal VC system would then consist two components: the recognition module and the synthesis module, where the two components perform the abovementiond respective actions. It is therefore essential for a successful VC model to find a speaker-independent intermediate feature space that purely reflects the linguistic contents. Such property is, however, hard to facilitate in VC, especially when an seq2seq model is employed, since only a parallel corpus is used. We argue that, with the help of external data and knowledge, this property may be easier to capture.

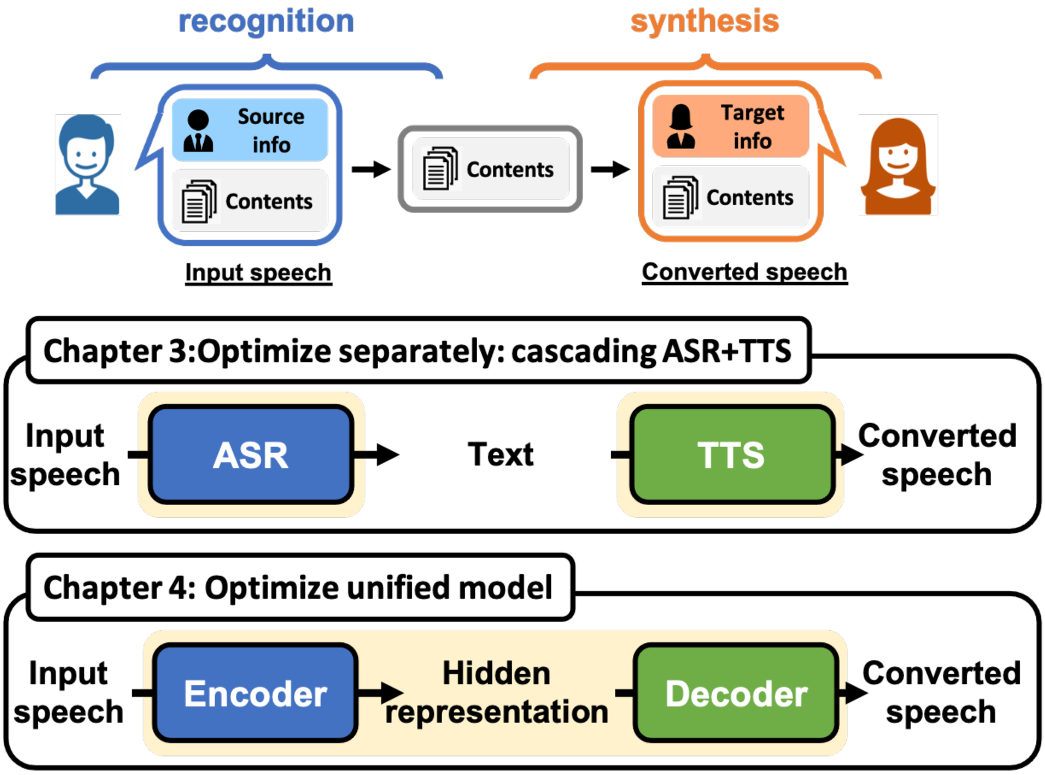


Figure 1.4: *An information perspective of voice conversion, and a summarization of the methods discussed in this thesis.*

1.3.2 Method 1: Concatenate separately optimized ASR and TTS models

Our first approach is a rather naive approach, which is to directly concatenate an ASR model and a TTS model, trained using separate, large-scale corpora in the respective fields. As mentioned in Section 1.1, ASR and TTS are two of the most active research fields in speech processing, and there are carefully curated datasets designed for the respective tasks, which are easier to collect compared to the parallel datasets required by VC. It is therefore expected that the use of the massive external corpora can boost the VC performance. On the other hand, from the model point of view, seq2seq modeling is naturely suitable for modeling the mapping function between speech and text considering the time resolution and temporal structure, thus modern ASR and TTS systems often employ seq2seq models. As a result, although the concept of concatenating ASR and TTS models itself is not novel, it is still worthwhile revisiting this method using state-of-the-art seq2seq models to form a seq2seq VC system as a whole.

To demonstrate the effectiveness of this simple method, we adopted this system as one of the baseline systems of the voice conversion challenge (VCC) 2020. The official listening test results revealed that the ASR+TTS method served as a strong and competitive baseline, ranking 2nd in terms of speaker similarity among the 30 participating teams.

As shown in Table 1.1, this method requires the ASR and TTS modules during both training and runtime conversion to generate and make use of the text. In addition, text label of the training data is required for the ASR and TTS training.

1.3.3 Method 2: Optimize one unified model

One obvious fallback of the cascade system described in Section 1.3.2 is that the ASR and TTS models are optimzied separately, and it is generally believed that an unified system would perform better. In the second method, we focus on optimizing one seq2seq model, and propose pretraining strategies utilizing ASR and TTS to alleviate the data deficiency problem. Considering that TTS and ASR both aim to find a mapping between text and speech, as the former tries to add speaker information to the source while the latter tries to remove, we suspect that the intermediate hidden representation spaces of these two tasks contain somewhat little speaker information, and serve as a suitable fit for VC. The ability of the pretraining technique to alleviate the

Table 1.1: *Comparison of the two methods proposed in this thesis.*

Method	Extra module Training	 Runtime	Text label of training data	Regularization
Method 1	✓	✓	✓	–
Method 2	✓	–	–	optional

data deficiency problem was confirmed in the experiments, as it was shown that models with pretraining could significantly improve several metrics including naturalness, speaker similarity and intelligibility.

As shown in Table 1.1, this method requires the ASR or TTS module during training, but only the final VC model is required during runtime conversion. The text label of the VC training data is not required. Regularization is an optional choice, as it can further improve the VC performance but not necessary.

1.4 Thesis overview

This thesis is organized as follows. In Chapter 2, we introduce the basics of seq2seq modeling in VC, including two model architectures, namely the rerurrent neural network (RNN)-based and Transformer-based models. In Chapter 3, we describe the implementation of the ASR+TTS framework, and the official evaluation results in VCC2020. In Chapter 4, the pretraining technique the transfers knowledge from ASR and TTS to an unified seq2seq VC modeling framework is presented. Finally, in Chapter 5, the contributions of this thesis are summarized and future work is discussed.

2 Sequence-to-Sequence Modeling for Text-to-Speech and Voice Conversion

This section provides descriptions on basics of seq2seq modeling for TTS and VC. Both falling in the category of speech synthesis, TTS and VC take different modalities as input but share a common goal of synthesizing speech waveform, as depicted in Figure 2.1. As a result, the model architecture is very similar, and one can easily make a few modifications to a TTS model to form a model for VC. Thus, we first provide a unified formulation of seq2seq modeling for speech synthesis, and some common components shared by the models. Then, we introduce two seq2seq VC model architectures that are used in this work. The first one is based on the recurrent neural network (RNN), which was used in the very first seq2seq models, as well as many seq2seq VC models [39,40,45,56–58,60]. The second one is the Transformer architecture [61], which has shown promising results in many speech processing tasks, including VC.

2.1 Unified seq2seq modeling for TTS and VC

Seq2seq models learn a mapping between a source feature sequence $X = \boldsymbol{x}_{1:n} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ and a target feature sequence $Y = \boldsymbol{y}_{1:m} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m)$ which are often of different length, i.e, $n \neq m$. In the case of TTS, the input can be a character or word

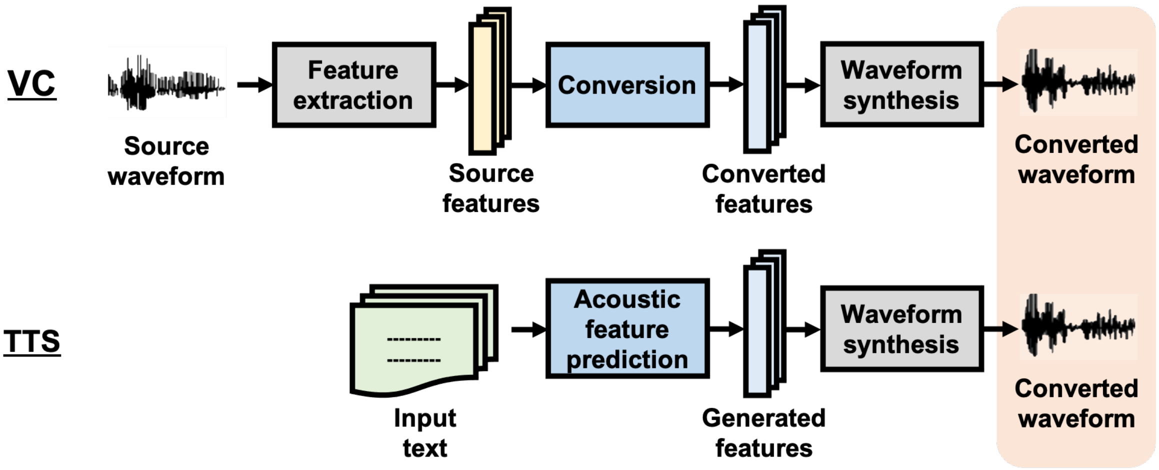


Figure 2.1: *Illustration of the input and output modalities of TTS and VC.*

sequence, and in the case of VC, the input can be an acoustic feature sequence such as a log mel-spectrogram. The output for both tasks is always an acoustic feature sequence. As with most seq2seq models, ours also has an encoder—decoder structure [36]. For instance, Figure 2.2 depicts the general structure of a seq2seq VC model.

The encoder (Enc) first maps the input feature sequence $\boldsymbol{x}_{1:n}$ into a sequence of hidden representations:

$$H = \boldsymbol{h}_{1:n} = \mathrm{Enc}(\boldsymbol{x}_{1:n}). \tag{2.1}$$

The decoder (Dec) is autoregressive, which means that when decoding the current output $\boldsymbol{y}_t$, in addition to the encoder output, i.e. the hidden representations $\boldsymbol{h}_{1:n}$, the previously generated features $\boldsymbol{y}_{1:t-1}$ are also considered:

$$\boldsymbol{y}_t = \mathrm{Dec}(\boldsymbol{h}_{1:n}, \boldsymbol{y}_{1:t-1}). \tag{2.2}$$

Some extra components and techniques are adopted in the seq2seq model to improve performance and stabilize training, most of which are inspired by the success of modern seq2seq TTS models [62, 63].

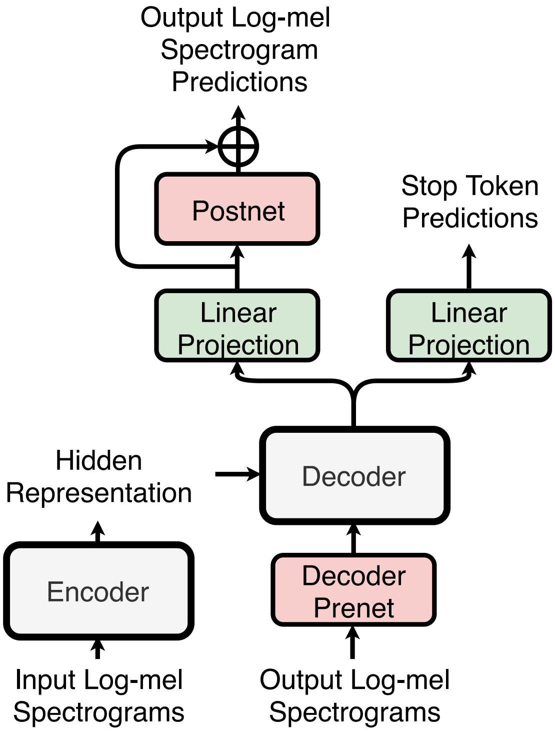


Figure 2.2: *Illustration of the unified seq2seq VC model architecture and the shared components.*

- A prenet containing 2 fully connected layers is added to the decoder, which serves as an information bottleneck essential for learning the autoregressive decoder.

- A linear projection layer is used to project the decoder output to have the desired dimension. To learn when to stop decoding, a separate linear projection layer is used to predict a stopping probability, which can be used with a threshold to decide when to stop decoding during inference

- To compensate for the missing context information in the autoregressive decoder, a five-layer CNN postnet is used to predict a residual that is added to the projected output.

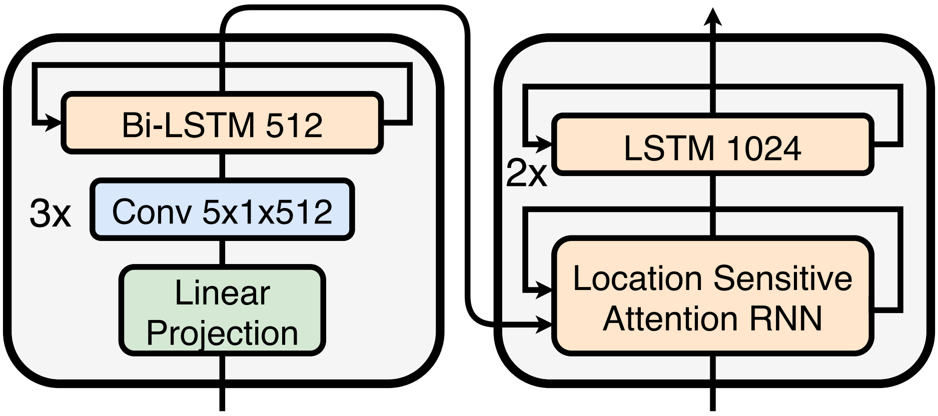


Figure 2.3: *RNN-based encoder and decoder.*

- Introducing the reduction factor r greatly helps speed up convergence and reduce training time and memory footprint. Specifically, at each decoding step, r non-overlapping frames are predicted. Since adjacent speech frames are often correlated, this technique allows the decoder to correctly model the interaction with the hidden representation sequence.

The training objectives include an L1 and L2 loss, in combination with a weighted binary cross-entropy loss on the stop token prediction and additional objectives that stablizes training. The whole network is composed of neural networks and optimized via backpropagation.

2.2 RNN-based model

Our RNN-based seq2seq VC model is based on the Tacotron2 TTS model [63] and resembles the work in [39]. The encoder first linearly projects the input log-mel spectrogram, followed by a stack of convolutional layers, batch normalization, and ReLU activations. The output of the final convolutional layer is then passed into a bi-directional

LSTM layer to generate the hidden representations.

For each decoder output step, an attention mechanism [37, 38] is used to attend to different positions of the hidden representation sequence. First, a context vector $\boldsymbol{c}_t$ is calculated as a weighted sum of $\boldsymbol{h}_{1:n}$, where the weight is represented using an attention probability vector $\boldsymbol{a}_t = (a_t^{(1)}, \cdots, a_t^{(n)})$. Each attention probability $a_t^{(k)}$ can be thought of as the importance of the hidden representation $\boldsymbol{h}_k$ at the current time step. As in Tacotron2, we adopt the location-sensitive attention [64], which takes cumulative attention weights from previous decoder time steps as an additional feature to encourage a forward consistency to prevent repeated or missed phonemes. The context vector is then concatenated with the prenet output and passed into a stacked uni-directional LSTM network to predict the r output frames. The above-mentioned procedure can be formulated as follows:

$$\boldsymbol{a}_t = \text{attention}(\boldsymbol{q}_{t-1}, \boldsymbol{h}_{1:n}), \tag{2.3}$$

$$\boldsymbol{c}_t = \sum_{k=1}^{n} a_t^{(n)} \boldsymbol{h}_k, \tag{2.4}$$

$$\boldsymbol{y}_t, \boldsymbol{q}_t = \text{Dec}(\boldsymbol{y}_{1:t-1}, \boldsymbol{q}_{t-1}, \boldsymbol{c}_t). \tag{2.5}$$

2.2.1 Guided Attention Loss

For seq2seq speech synthesis models, the attention alignment is usually monotonic and linear, so a guided attention loss that encourages the attention matrix to be diagonal can speed up attention learning and convergence [39,65]. The assumption is that the i-th element in the input feature sequence progresses nearly linearly with respect to the j-th element of the output feature sequence, i.e., $i \sim \alpha j$, where $\alpha \sim \frac{n}{m}$. Therefore, the attention matrix $A = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_m]$ should be a nearly diagonal. We may therefore define a penalty matrix G:

$$g_{i,j} = 1 - \exp\left\{ \frac{-(\frac{i}{n} - \frac{j}{m})^2}{2\sigma_g^2} \right\}, \tag{2.6}$$

where σ_g controls how close A is to a diagonal matrix. The guided attention loss $\mathcal{L}_{\text{ga}}$ is then defined as

$$\mathcal{L}_{\text{ga}} = \lambda_{\text{ga}} ||G \odot A||_1, \tag{2.7}$$

where $\odot$ indicates an element-wise product and λ_{ga} is the weight for the guided attention loss.

2.2.2 Context Preservation Loss

In [39], in addition to the guided attention loss, a context preservation loss was further applied to maintain linguistic consistency after conversion. Specifically, to encourage the source encoder generate meaningful hidden representations, we introduce two additional networks as a context preservation mechanism: a source decoder SrcDec for reconstructing the source feature sequence from the hidden representations, and a target decoder TarDec for predicting the target feature sequence from the context vectors, $C = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_m]$:

$$\tilde{X} = \text{SrcDec}(H), \tag{2.8}$$

$$\tilde{Y} = \text{TarDec}(C). \tag{2.9}$$

The context preservation loss is then defined as:

$$\mathcal{L}_{\text{cp}} = \lambda_{\text{cp}}(||\tilde{X} - X||_1 + ||\tilde{Y} - Y||_1), \tag{2.10}$$

where λ_{cp} is the weight for the context preservation loss.

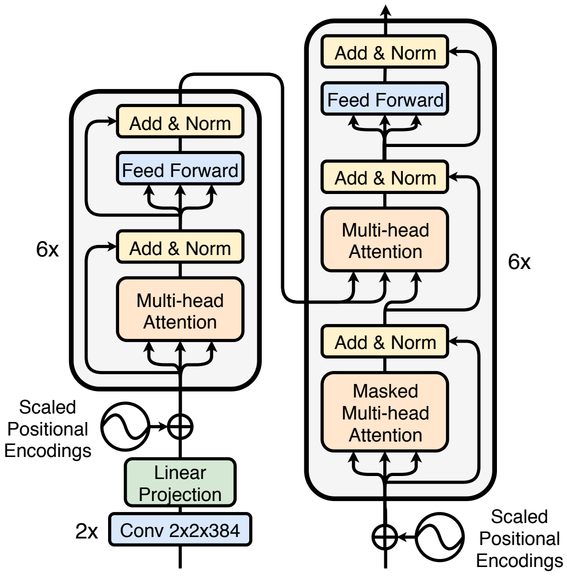


Figure 2.4: *Transformer-based encoder and decoder.*

2.3 Transformer-based model

Our Transformer-based seq2seq VC model, which we refer to as the Voice Transformer Network (VTN), is based on the Transformer architecture [61], which was originally designed for machine translation but also widely applied to other sequential modeling problems. There are several core components of the Transformer:

Multi-head attention (MHA) sublayer. The MHA layer is defined as:

$$\mathrm{MHA}(Q, K, V) = [\mathrm{head}_1, \cdots, \mathrm{head}_h]W^O, \tag{2.11}$$

$$\mathrm{head}_i = \mathrm{Att}(QW_i^Q, KW_i^K, VW_i^V), \tag{2.12}$$

where Q, K and V denote the input matrices that, following [61], are referred to as the query, key and value, respectively. MHA uses h different, learned linear projections W^Q, W^K, W^V to map the inputs to different *heads*, and then perform the Att operation in parallel. The outputs from all heads are concatenated and projected with W^O. As in [61], the Att operation is implemented scaled dot-product attention is used:

$$\text{Att}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_{att}}})V, \tag{2.13}$$

where d_{att} is the attention dimension.

Position-wise feed-forward network (FFN) layer. The FFN layer is defined as:

$$FFN(\boldsymbol{x}) = \max(0, \boldsymbol{x}W_1 + b_1)W_2 + b_2, \tag{2.14}$$

which is independently applied at each time step (position) with different parameters from layer to layer.

Layer normalization and residual connection. Around either of the above-mentioned sublayers, a residual connection followed by layer normalization [66] is employed. For input X of a sublayer, the output is given as:

$$\text{LayerNorm}(X + \text{Sublayer}(X)). \tag{2.15}$$

Due to the residual connections, all sublayers have the same output dimension d_{model}.

Scaled positional encoding (SPE). In the original Transformer [61], since no recurrent relation is employed in the Transformer, to let the model be aware of information about the relative or absolute position of each element, the triangular (sinusoidal) positional encoding (PE) [67] is added to the inputs to the encoder and decoder. In this work, we adopt the SPE [68], which is a generalized version of the original PE that scales the encodings with a trainable weight α, so that they can adaptively fit the scales of the encoder and the decoder:

$$\text{SPE}(t) = \begin{cases} \alpha \cdot \sin(\frac{t}{10000^{\frac{2t}{d_{\text{model}}}}}), & \text{if } t \text{ is even,} \\ \alpha \cdot \cos(\frac{t}{10000^{\frac{2t}{d_{\text{model}}}}}), & \text{if } t \text{ is odd.} \end{cases} \tag{2.16}$$

The encoder we adopt in this work resembles the one in [69]. First, the input acoustic feature sequence is downsampled in the time and frequency axes by a fraction of 4 using two convolutional layers with stride 2×2. While the reduction of the memory footprint is a clear benefit, a hidden representation with a low sampling rate can not only speed up attention learning convergence due to easier attention calculation at each decoding step but also approximates phoneme-level or even character-level linguistic contents [40]. After linearly projecting to d_{model}-dimensions and adding the SPE, L identical encoder layers are stacked to form the core of the encoder. Each encoder layer consists of an MHA sublayer and an FFN sublayer, followed by a residual connection and layer normalization. The MHA layers in the encoder are *self-attention* layers since the queries, keys, and values are all from the output from the previous layer.

The decoder in this work is composed of the same number of L identical decoder layers as in the encoder. In each decoder layer, the first sublayer is the so-called *masked* self-attention MHA sublayer, where a mask is utilized such that at time step t, only vectors with time index up to and including t can be accessed. This preserves the autoregressive property of the model. Then, an MHA sublayer uses the outputs from the previous layer as queries and H as the keys and values, which ensembles the encoder—decoder attention in 2.2. Finally, an FFN sublayer is used, as in the encoder. Again, all sublayers are wrapped with a residual connection and layer normalization.

In addition to the L1, L2, and weighted binary cross-entropy losses, the guided attention loss is also applied. As pointed out in [68], in Transformer-based speech synthesis, not all attention heads demonstrate diagonal alignments, so following [70,71], the guided attention loss is applied to partial heads in partial decoder layers.

3 Cascading ASR and TTS: Baseline System for the Voice Conversion Challenge 2020

3.1 Introduction

The aim of the voice conversion challenge (VCC)[1] is to better understand different VC techniques built on a freely-available common dataset to look at a common goal and to share views about unsolved problems and challenges faced by current VC techniques. The challenges focused on *speaker conversion*, where VC models are built to automatically transform the voice identity. In the third version, VCC2020 [44], two new tasks are considered. The first task is *semiparallel* VC within the same language, where only a small subset of the training set is parallel with the rest being nonparallel. The second task is *cross-lingual* VC, where the training set of the source speaker is different from that uttered by the target speaker in language and content, thus nonparallel in nature. In conversion, the source speaker's voice in the source language is converted as if it was uttered by the target speaker while keeping linguistic contents unchanged.

It would be worth discussing two important factors when designing a VC system: data and model. First, from the data point of view, in either of the VCC2020 tasks,

[1] `http://www.vc-challenge.org/`

techniques for dealing with nonparallel data need to be developed. In the literature, a promising paradigm for nonparallel VC is through a recognition-synthesis framework. The idea is to first extract from the source speech the linguistic contents, followed by blending with the target speaker characteristics to generate the converted speech. Methods implementing this framework can be divided according to the type of linguistic representation. The first type encodes representations with an ASR model, where a popular choice is the phonetic posteriorgram (PPG) [55,72]. A synthesis model is then trained to generate the voice of the target speaker. The second type usually employs an autoencoder-like model that estimates the recognizer and synthesizer simultaneously by implicitly factorizing the linguistic and speaker representations [73–77].

From the model point of view, we have witnessed how seq2seq models [36] change the game in many research fields in only half a decade, and speech processing is no exception. Its application in VC is especially attractive since that compared to conventional frame-based methods that perform conversion frame-by-frame, seq2seq models can implicitly learn the complex alignment and relationship between the source and target sequences to generate outputs of various lengths. It is therefore a natural choice to convert prosody including the speaking rate and F0 contour, which is closely related to speaker characteristics. As a result, seq2seq based VC has been a promising approach in terms of conversion similarity [39–41,58].

In this paper, we describe the seq2seq baseline system for the VCC2020. Our system is a cascade of seq2seq-based ASR and TTS models, which we will refer to as ASR+TTS. A suitable baseline system should meet the following requirements:

- The system should be a simple and easy-to-use starting ground for newcomers to base their work on.

- The system should be an open-source project made publicly available to benefit

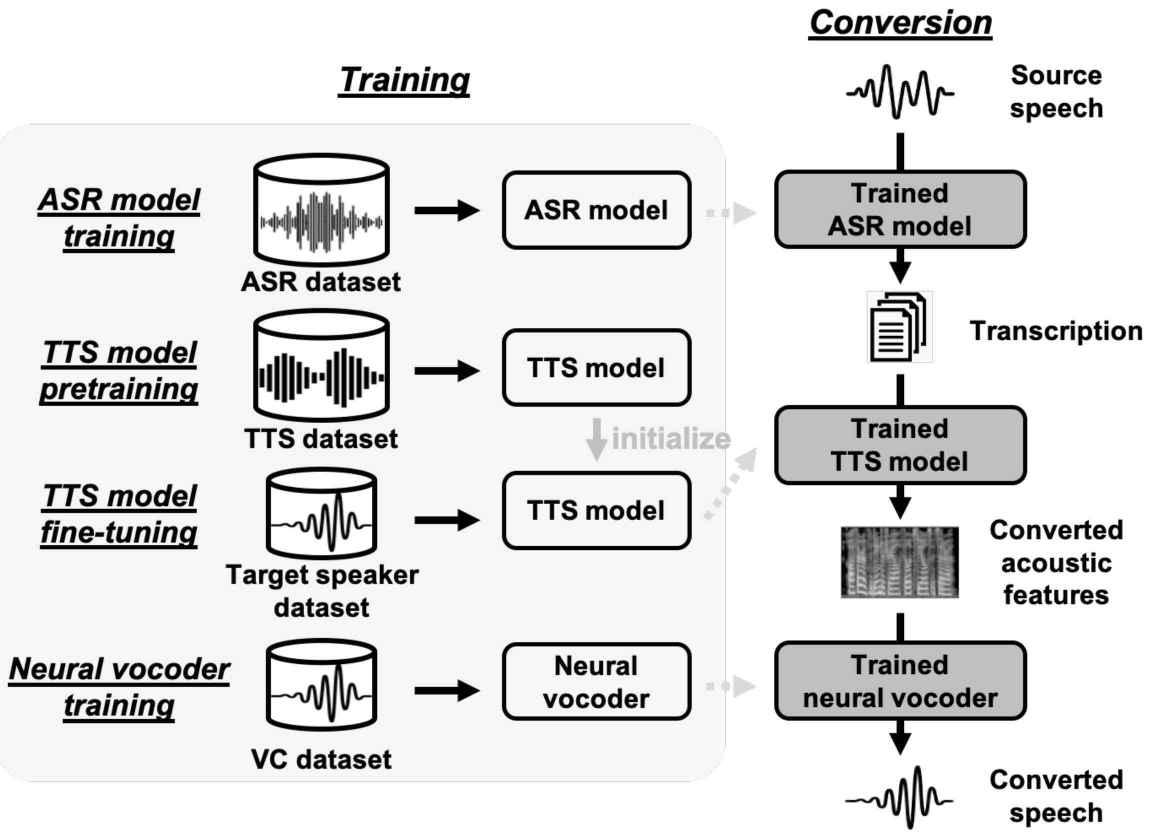


Figure 3.1: *The training and conversion processes of the* ASR+TTS *method.*

potential future researchers.

- The system should serve as a competitive benchmark.

With these goals in mind, we implemented the system using ESPnet, a well-developed open-source end-to-end (E2E) speech processing toolkit [71, 78], and made as much use of publicly available datasets as possible. Although it is generally believed that simply cascading systems to perform a certain task is inferior to an end-to-end model, benefitting from recent advances in ASR and TTS, as well as efforts such as implementation and hyperparameter tuning which are dedicated by the open-source community, we will show that our system is not only easy to use but serves as a strong competing system in the VCC2020.

3.2 System Overview

A naive approach for VC is a cascade of an ASR model and a TTS model. Although this method is not new, by revisiting this method using seq2seq models, we can model the prosody such as pitch, duration, and speaking rate, which is usually not well considered in the literature. Conceptually speaking, the ASR model acts like a speaker normalizer that first normalizes the input speech such that attributes of the source speaker are filtered out and only the linguistic content remains. Then, the TTS model functions to add speaker information to the recognition result so that the converted speech sounds like the target speaker.

Our system, as depicted in Figure 3.1, consists of three modules: a speaker-independent ASR model, a separate speaker-dependent TTS model for each target speaker, and a neural vocoder that synthesizes the final speech waveform.

ASR model. ASR models are usually trained with a multi-speaker dataset, thus speaker-independent in nature. For both tasks 1 and 2, the source speech is always English, so an English transcription is first obtained using the ASR model.

TTS model. In the TTS literature, it is a common practice to train in a speaker-dependent manner rather than training speaker-independently since the former usually outperform the latter. However, the size of the training set of each target speaker in VCC2020 is too limited for seq2seq TTS learning. In light of this, we employ a pretraining-finetuning scheme that first pretrains on large TTS datasets followed by fine-tuning on the limited target speaker dataset [79] . This allows us to successfully train on even approximately 5 minutes of data.

Neural vocoder. In recent years, neural waveform generation modules (also known as vocoders) have brought significant improvement to VC. In this work, we use the Parallel WaveGAN (PWG) [80], since it enables high-quality, real-time waveform generation. An open-source implementation[2] is adopted and we integrated it with ESPnet.

Our implementation was built upon the E2E speech processing toolkit ESPnet [71, 78], which provides various useful utility functions and properly tuned pretrained models.

3.3 ASR Implementation

3.3.1 Data

Since the input is always English, we used the Librispeech dataset [81], which contained 960 hours of English speech data from over 2000 speakers.

3.3.2 Model

The backbone of the ASR model was the Transformer [61, 69, 82]. The model was trained in an end-to-end fashion using a hybrid CTC/attention loss [83], and a recurrent neural network based language model (RNNLM) was used for decoding. We directly used a pretrained model (including the RNNLM) provided by ESPnet.

3.4 TTS Implementation

We are faced with a harder challenge in implementing the TTS model. In task 2, the input language is different from the languages of the training data. In other words,

[2] `https://github.com/kan-bayashi/ParallelWaveGAN`

Table 3.1: *The TTS training datasets in task 2. "phn" and "char" stand for phoneme and character, respectively.*

Lang.	Dataset	Spkrs	Hours	Input
Eng.	M-AILABS [84]	2	32	phn or char
Ger.	M-AILABS [84]	5	190	char
Fin.	CSS10 [85]	1	10	char
Man.	CSMSC [86]	1	12	pinyin

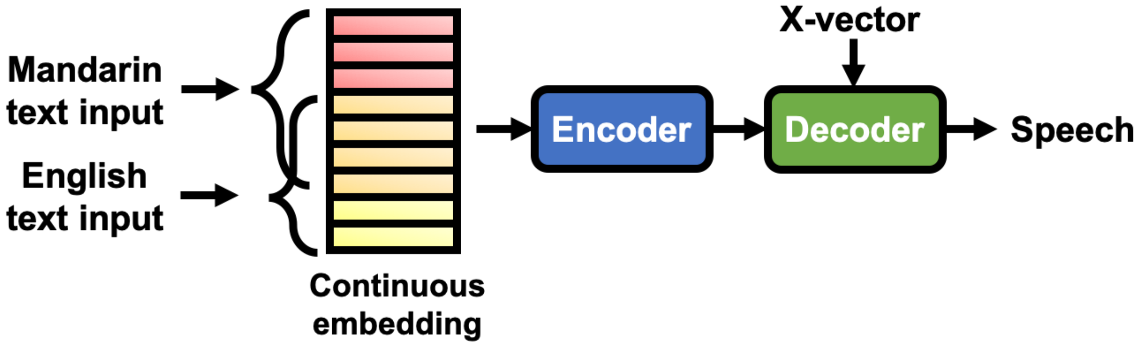


Figure 3.2: *Illustration of the bilingual TTS used in task 2.*

the TTS model needs to lean the voice of an unseen language. This is sometimes referred to as cross-lingual voice cloning [87, 88]. As there has not been a standard, promising protocol especially when only five minutes of training data is available, we adopt a simple method that constructs x-vector [89] based, bilingual TTS models by pretraining with corpora of English and the target language and finetuning with the target language.

3.4.1 Data

The target language for task 1 is English, so for pretraining, we used the multi-speaker LibriTTS [90] dataset, which contained around 250 hours of English data from over 2000 speakers. In task 2, the target languages are German, Finnish, and Mandarin. Considering the open-source ability, we wish to avoid using commercial or private datasets. Unfortunately, under such constraint, there is not much choice, and the available datasets at the time we developed the system were large but contained only data from a single speaker or a few speakers, as shown in Table 3.1. Although it has been shown that combining imbalanced multi-speaker datasets improves performance [91], this effect remains unknown in the cross-lingual setting. To this end, for the English data, we decided to use not the LibriTTS dataset which has many speakers yet a small amount of data per speaker, but the M-AILABS dataset [84], which has a large amount of data from a few speakers only. Finally, since the task 2 datasets were of different sampling rates, we doswnsampled all task 2 data to 16 kHz. As for the x-vector extractor, the Kaldi toolkit was used and the model was pretrained on VoxCeleb [92].

3.4.2 Model

We used an x-vector [89] based multi-speaker TTS model [93] with a Transformer backbone [68]. The input was a linguistic representation sequence, and the output was the mel filterbank sequence extracted from the (optionally downsampled) waveform. In task 1, since the input is always English, the model simply takes English characters an input.

However, in task 2, it is nontrivial to decide the input representation since it is often language-dependent. For example, there is no overlap in the text representation between Mandarin and English [87]. When we finetune a pretrained model for a Mandarin speaker, since the Mandarin corpus does not contain English words, the model has no clue how the target speaker pronounce English words. This mismatch may cause quality degradation. Below, we describe how we alleviate this issue.

We used a shared input embedding space when training the bilingual TTS model. In neural TTS, the input embedding look-up table is a projection from discrete input symbols to continuous representation and is trained with the rest of the model by backpropagation. It is useful in that the model can implicitly learn how to pronounce each input token, such that different tokens with a similar pronunciation can have a similar embedding. The assumption here is that there is an overlap between the input representations of the two languages. For example, if we train a Mandarin/English TTS model, the ”ah” phoneme in English and ”a” pinyin representation may have similar embeddings. As a result, even if only ”a” is seen during training, by learning how the target speaker pronounces such vowel, the model may still know how to pronounce ”ah”.

For the Mandarin/English TTS, we used phonemes and pinyin as input, while for the Finnish/English and German/English TTS, we used characters as input. In the finetuning stage, the parameters are updated using the training utterances of the target speaker, except that the embedding lookup table in Figure 3.2 is fixed.

3.5 Neural Vocoder Implementation

The PWG had a non-autoregressive (non-AR) WaveNet-like architecture and was trained by jointly optimizing a multi-resolution spectrogram loss and a waveform adversarial loss [80]. The input was mel filterbank and the output was raw waveform.

For each task, we trained a separate PWG using the training data from all available speakers. In other words, data of 8 and 10 speakers were used to train PWGs for tasks 1 and 2, respectively. Notably, in task 2, although the mel filterbanks were extracted from 16kHz waveform as mentioned in Sections 3.4.1 and 3.4.2, we still map them to 24kHz waveform in training, as the quality degradation from such mismatch has shown to be acceptable [90]. Although the finetuning technique of neural vocoders has been proven to be useful in VC [34, 72], techniques for fine-tuning non-AR vocoders have not been well investigated, so we leave this for future work.

3.6 Challenge Results

3.6.1 VCC2020 Dataset

The VCC2020 database had two male and two female English speakers as the source speakers. For task 1, two male and two female English speakers were chosen as the target speakers, and one male and one female for each of Finnish, German, and Mandarin in task 2. Each of the source and target speakers has a training set of 70 sentences, which is around 5 minutes of speech data. Note that in task 1, the target and source speakers have 20 parallel sentences, where the rest 50 sentences are different. The test sentences for evaluation are shared for tasks 1 and 2 with a number of 25.

3.6.2 Evaluation protocol

The VCC2020 organizing committee conducted a large-scale subjective test on all submitted systems for both tasks 1 and 2. The evaluations included naturalness and similarity tests. In the naturalness test, a five-point mean opinion score (MOS) test

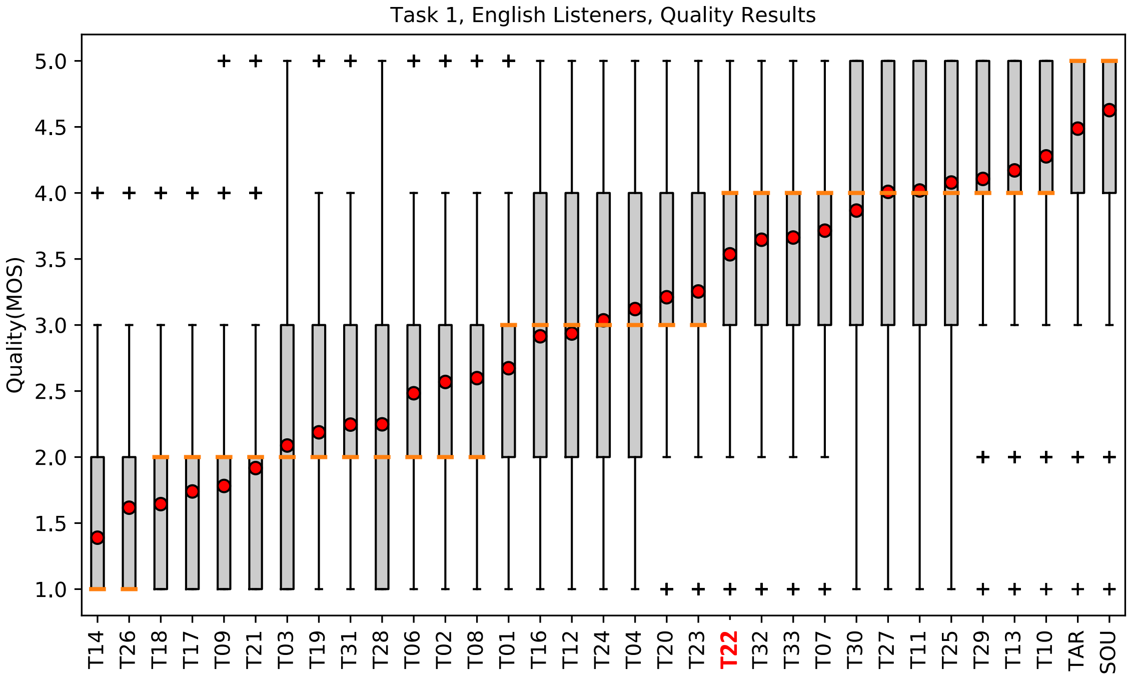


Figure 3.3: *Naturalness results for task 1.*

was adopted, where listeners were asked to rate the naturalness of each speech clip from 1 to 5. In the similarity test, listeners were presented with a converted and a ground truth target utterance, and they were asked to decide whether or not the two utterances were spoken by the same person on a four-point scale[3].

3.6.3 Task 1 Results

Figures 3.3 and 3.4 show the overall results for task 1. For naturalness, our system received a MOS score of about 3.5, which ranks 11 out of all the 31 submitted systems in task 1. This shows that, as many systems are specifically designed for VC, sim-

[3] Although the official report contained results from Japanese and English listeners, here we only report results of English listeners since the two listener groups share a similar tendency.

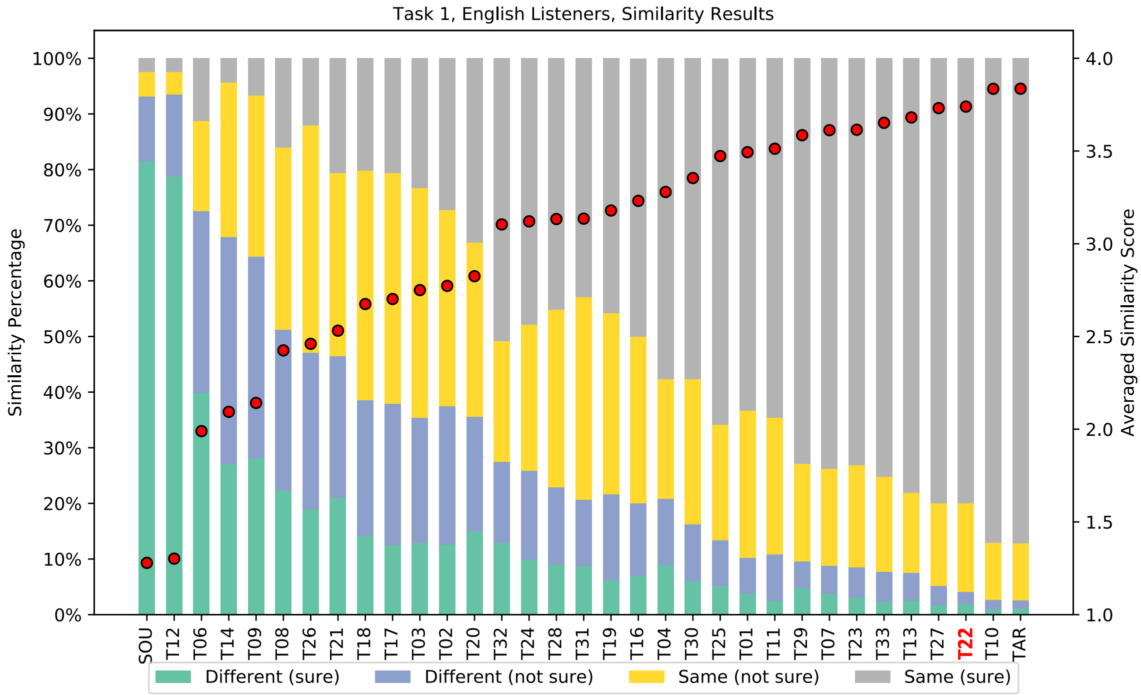


Figure 3.4: *Similarity results for task 1.*

ply combining state-of-the-art ASR and TTS systems can already achieve competitive results, thanks to the well-developed technologies in the two research fields. The performance gap between our system and the superior teams may come from the difficulty of finetuning the TTS model with only 70 utterances. As for similarity, our system had a similarity score around 90%, which means that about 90% of the converted utterances were considered spoken by the same target speaker by the participants. This made our system rank second among all teams, which serves as strong evidence of the superiority of seq2seq models when it comes to converting speaker identity.

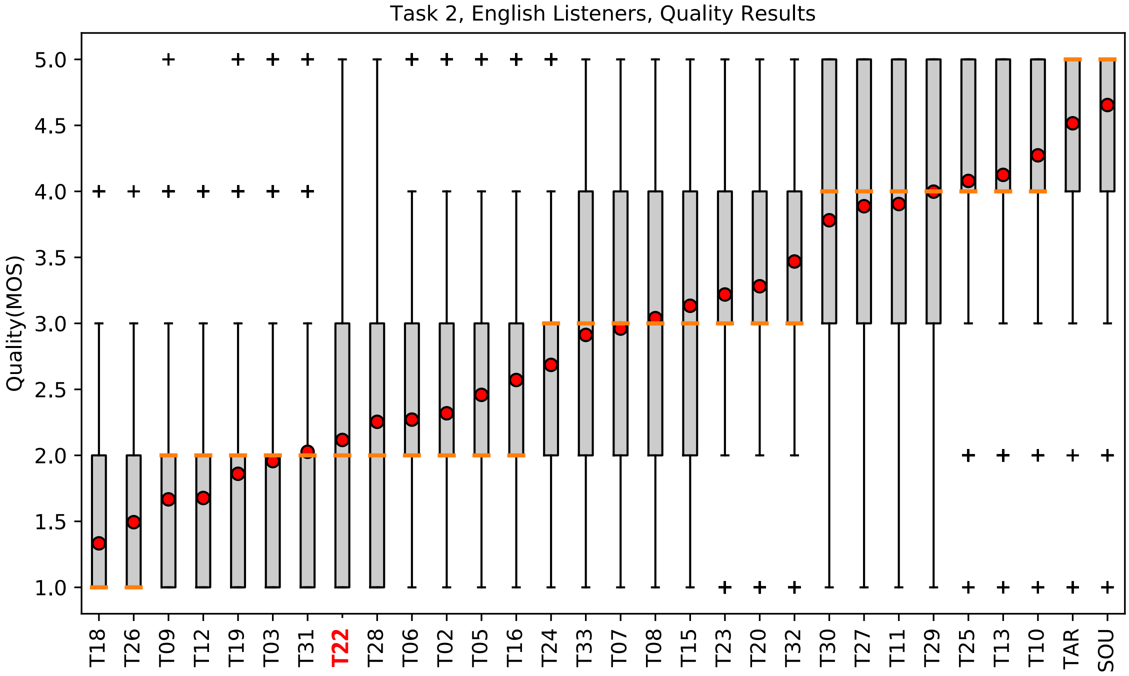


Figure 3.5: *Naturalness results for task 2.*

3.6.4 Task 2 Results

Figures 3.5 and 3.6 show the overall results for task 2. For naturalness, our system had a MOS score of about 2.0, ranking 21 out of all the 28 submitted systems in task 2, which is a lot worse than the performance in task 1. On the other hand, our system ranked 9 among the 28 teams in the similarity test. Looking at these two results, it can be inferred that our system can still well capture the speaker characteristics thanks to the power of seq2seq modeling, but suffer from a severe quality degradation. This is possibly owing to the limited training data and the lack of pretraining data, as well as the difficulty of handling the cross-lingual data using the overly-simple TTS model we implemented.

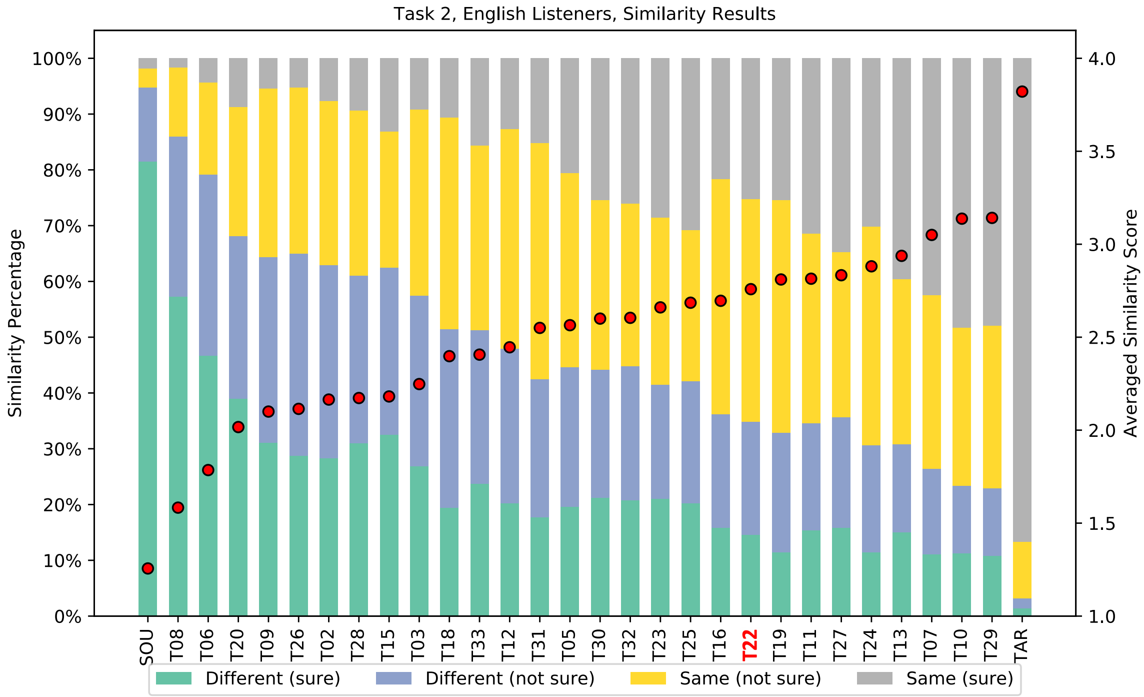


Figure 3.6: *Similarity results for task 2.*

3.7 Analysis on Linguistic Contents

A potential threat of the cascading paradigm is that error in early stages might propagate to downstream models. In our proposed method, the recognition failure in the first ASR stage might harm the linguistic consistency in VC. We examine this phenomena by measuring the intelligibility with an off-the-shelf Transformer-ASR model trained on LibriSpeech, which is provided in ESPnet.

Table 3.2 shows the ASR results. First, the error rates on the input source speech were not severe as they are similar to that on the test set of LibriSpeech. However, the scores of the converted speech are much worse, indicating that the imperfect TTS modeling is the main cause of intelligibility degradation. We also observe that the error rates of task 2 are much higher than that of task 1, which is consistent with the results

Table 3.2: *Character/word error rates (CER/WER) (%) calculated using a pretrained ASR model. The scores are averaged over all target speakers. Note that these results are not from the official objective evaluation.*

	Input		Task 1		Task 2	
Source	CER	WER	CER	WER	CER	WER
SEF1	2.9	6.5	12.1	22.1	19.9	34.3
SEF2	1.4	3.7	12.6	22.7	21.4	36.2
SEM1	0.2	0.9	14.2	20.1	20.3	36.8
SEM2	2.9	7.5	18.5	30.9	22.7	38.0

in Section 3.6.4.

3.8 Conclusion and Discussion

This paper described the seq2seq baseline system of the VCC2020, including the intuition, system design, training datasets, and results. Built upon the E2E, seq2seq framework, our ASR+TTS baseline served as a simple starting point and a benchmark for participants. Subjective evaluation results released by the organizing committee showed that our system is a strong baseline in terms of conversion similarity, confirming the effectiveness of seq2seq modeling. The results also demonstrate the naive yet promising power of combining state-of-the-art ASR and TTS models. Yet, there is still much room for improvement, and below we discuss several possible directions that might be addressed in an advanced version.

Enhance the pretraining data. As stated in Section 3.4.1, there was not much choice for pretraining data in task 2 under the open-source constraint. Using a multi-speaker pretraining dataset as in task 1 might improve the performance. Also, using datasets with a higher sampling rate can also improve the quality of the vocoder.

Utilize linguistic knowledge. One principal of E2E learning to use as less domain-specific knowledge as possible, That is to say, the system performance is expected to be improved when such knowledge is utilized. For example, as reported in [87], using phoneme inputs can greatly improve multi-lingual TTS systems, but we could not do so in task 2 due to the unfamiliarity with target languages such as Finnish and German.

Select an advanced multi-speaker TTS model. The multi-speaker TTS model [93] we adopted was a rather naive one, and a more state-of-the-art model like [94] might improve the performance.

Improve the neural vocoder. We adopted a non-AR neural vocoder for fast generation, but it is generally believed that AR ones are still superior. As this is a popular research field, it is expected that real-time neural vocoders maintaining the output quality will soon be developed. Also, finetuning the vocoders can further improve the performance, as stated in Section 3.5.

4 Pretraining for Parallel, One-to-one Sequence-to-sequence Voice Conversion

4.1 Introduction

Different from the cascade system discussed in chapter 3 where the ASR and TTS models are optimzied separately, in this chapter, we focus on optimizing one seq2seq model in a parallel, one-to-one setting. That is to say, we assume that the source speaker and the target speaker during the training and conversion phases are identical, and we have access to a parallel corpus which contains pairs of utterances of the same linguistic contents from the source and the target speakers. To tackle the data defeciency problem, we propose a pretraining strategy utilizing ASR and TTS.

Model pretraining, as one popular realization of transfer learning, has been regaining attention in recent years. This concept is usually realized by learning universal, high-level feature representations. In the field of computer vision, *supervised* pretraining (e.g. ImageNet classification [95–97]) followed by fine-tuning on tasks with less training data (e.g. object detection [98–100], segmentation [101, 102] and style transfer [103, 104]) often leads to state-of-the-art results. On the other hand, many natural language understanding (NLU) tasks learn rich representation through an *self-supervised* language model objective [105–109], which have also been shown to boost performance.

In speech processing, early applications of pretraining deep neural networks mainly lied in ASR, with the main goal of speeding up optimization and reducing generalization error [110, 111]. In recent years, inspired by the breakthrough in NLU, unsupervised or self-supervised speech representation learning utilizing massive, untranscribed speech data has become a popular research topic. As language modeling objectives have been widely employed for pretraining in NLU, finding a universally effective objective is still an active research area. Various objectives have been proposed, such as autoencoding [112–114] sometimes with an autoregressive model [115, 116] or contrastive learning [117–120]. Nonetheless, different pretraining objectives lead to different representations, and an effective objective for VC is still unclear.

We propose a novel yet simple pretraining technique to transfer knowledge from two speech processing tasks, namely TTS and ASR. We refer to them as *TTS-oriented pretraining* and *ASR-oriented pretraining*, respectively. In recent years, ASR and TTS systems based on neural seq2seq models have enjoyed great success owing to the vast large-scale corpus contributed by the community. We argue that lying at the core of these models is the ability to generate effective intermediate representations, which facilitates correct attention learning that bridges the encoder and the decoder. Nonetheless, we choose these two tasks not only because they are two of the most active research fields in speech processing, but because these two tasks, by definition, are suitable sources of transfer for VC.

We first provide a unified, intuitive explanation from an information perspective, as depicted in Figure 4.1. Roughly speaking, speech consists of the linguistic contents and the speaker identity. The goal of VC is to remove the source speaker information from the source speech, and then inject the identity of the target speaker. Thus, a

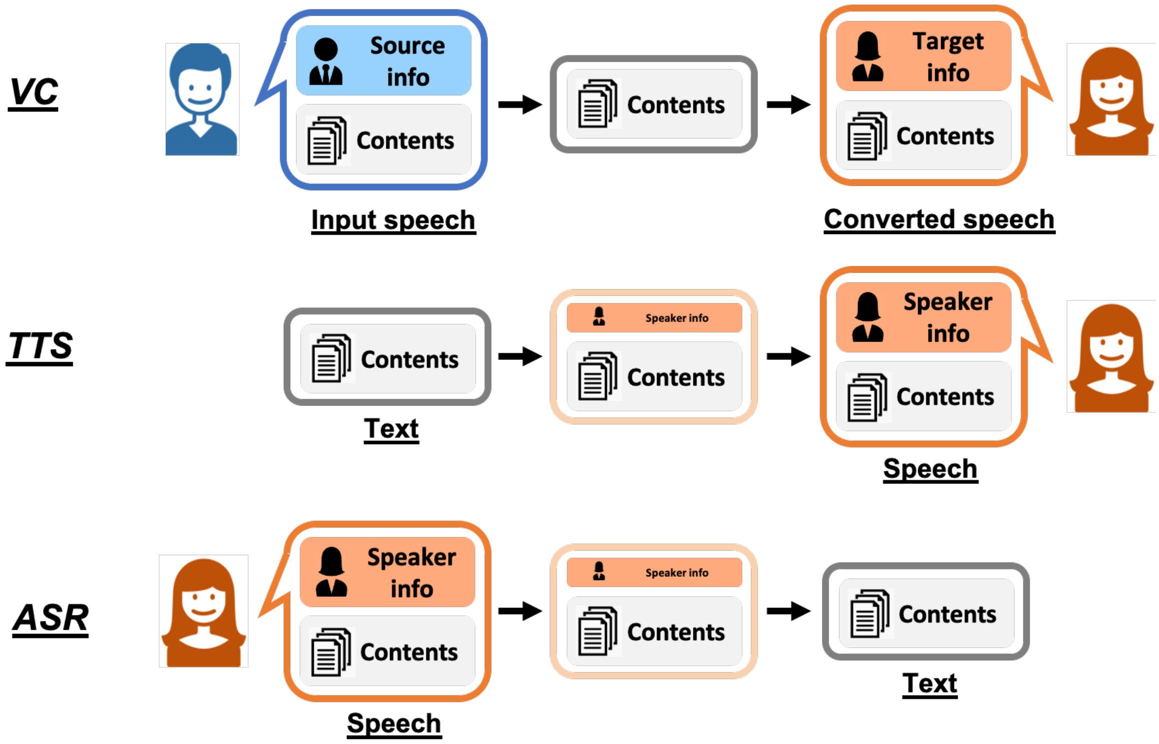


Figure 4.1: *Illustration of the relationship of VC, TTS and ASR in an information perspective.*

speaker-free intermediate feature space would be essential for a successful VC model, which is hard to facilitate given only a parallel corpus. On the other hand, TTS and ASR both aim to find a mapping between text and speech, as the former tries to add speaker information to the source while the latter tries to remove. We therefore suspect that the intermediate hidden representation spaces of these two tasks contain somewhat little speaker information, and serve as a suitable fit for VC.

Our method enjoys several advantages. First, our method relies on *supervised* pre-training with well-defined speech processing objectives. As we adopt popular speech processing tasks, large scale datasets can be assumed easily accessible thanks to the vastly growing community. Also, our method is flexible in that it needs neither text label of the VC data nor carefully designed regularization methods, yet can still achieve great data efficiency. Finally, it is expected that performance of pretraining would benefit from the rapid development of state-of-the-art models, thus improving the quality of the downstream VC task.

Our contributions are as follows:

- We propose TTS-oriented and ASR-oriented pretraining for seq2seq VC.
- We examine, through systematical objective and subjective evaluations, the TTS and ASR tasks for pretraining in seq2seq VC. Results show that they are both effective with sufficient data while only TTS pretraining maintains robust against the reduction of data.
- We visualize the hidden representation spaces of the learned models using different pretraining tasks and how they relate to the performance.
- We compare two different model architectures for seq2seq VC: recurrent neural networks (RNNs) and Transformers, and we show the supremacy of the latter over the former, which is consistent with the finding in most speech processing tasks [70].

4.2 Method

In seq2seq models for speech applications, effective intermediate representations can facilitate correct attention learning that bridges the encoder and the decoder, thus crucial to success. By the definition of VC, it is natural to try to encode the linguistic contents of the source speech into the hidden representations so that they can be maintained. Thus, we conjecture that the core ability of successful seq2seq VC models

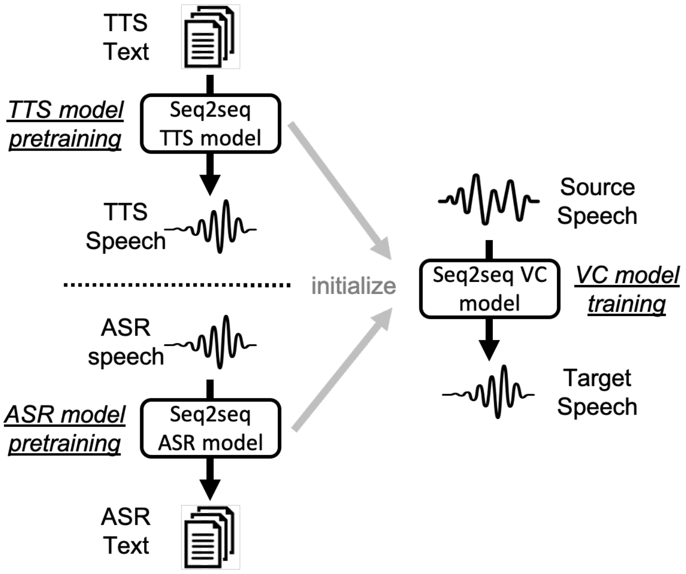


Figure 4.2: *Illustration of the concept of pretraining from seq2seq TTS or ASR to seq2seq VC.*

is to generate and utilize high-fidelity hidden representations.

In theory, both TTS and ASR tasks aim to find a mapping between two modalities: speech and text. As speech signals contain all essential linguistic information, the hidden representation spaces induced by these two tasks should lie in the middle of the spectrum between speech and text. Thus, we conjecture that such space is desirable for seq2seq VC models, thus suitable for pretraining.

In this work, we extend the TTS-oriented pretraining technique previously proposed in [41] to both TTS and ASR, as depicted Figure 4.2. Specifically, a two-stage training procedure is employed: in the first pretraining stage, a large-scale corpus is used to learn the initial seq2seq model parameters as a prior; then, in the second stage, the seq2seq VC model is initialized with the pretrained model parameters and trained with a relatively smaller VC dataset. The goal of this pretraining procedure is to provide fast, sample-efficient VC model learning, thus reducing the data size requirement and

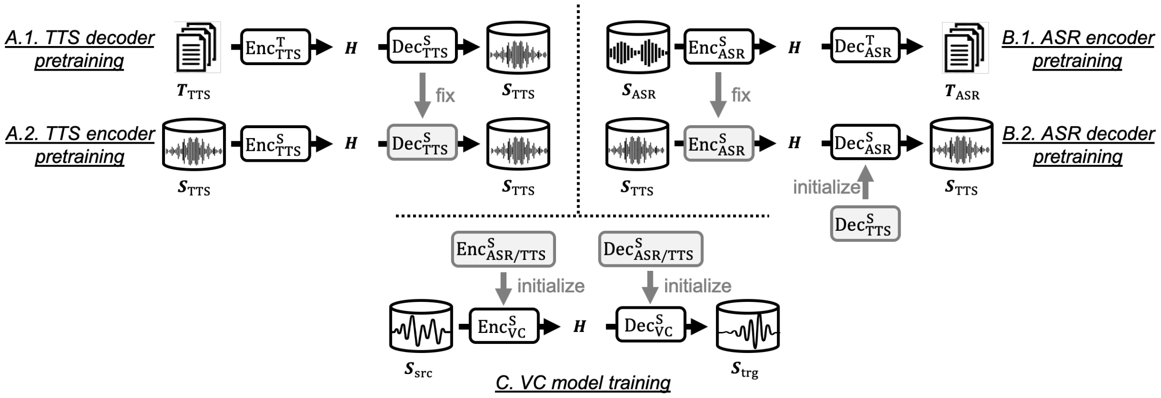

A.1. TTS decoder pretraining
T_{TTS}
$\text{Enc}^{\text{T}}_{\text{TTS}}$
H
$\text{Dec}^{\text{S}}_{\text{TTS}}$
S_{TTS}
fix
A.2. TTS encoder pretraining
S_{TTS}
$\text{Enc}^{\text{S}}_{\text{TTS}}$
H
$\text{Dec}^{\text{S}}_{\text{TTS}}$
S_{TTS}
S_{ASR}
$\text{Enc}^{\text{S}}_{\text{ASR}}$
H
$\text{Dec}^{\text{T}}_{\text{ASR}}$
T_{ASR}
B.1. ASR encoder pretraining
fix
S_{TTS}
$\text{Enc}^{\text{S}}_{\text{ASR}}$
H
$\text{Dec}^{\text{S}}_{\text{ASR}}$
S_{TTS}
B.2. ASR decoder pretraining
initialize
$\text{Dec}^{\text{S}}_{\text{TTS}}$
$\text{Enc}^{\text{S}}_{\text{ASR/TTS}}$
$\text{Dec}^{\text{S}}_{\text{ASR/TTS}}$
initialize
initialize
S_{src}
$\text{Enc}^{\text{S}}_{\text{VC}}$
H
$\text{Dec}^{\text{S}}_{\text{VC}}$
S_{trg}
C. VC model training

Figure 4.3: *Diagram of the pretraining procedures for TTS and ASR. Top left: TTS-oriented pretraining. Top right: ASR-oriented pretraining. Bottom: VC model training.*

training time. In addition, this setup is highly flexible in that we do not require any of the speakers to be the same, nor any of the sentences between the pretraining corpus and the VC dataset to be parallel.

Let the parallel VC dataset be $\boldsymbol{D}_{\text{VC}} = \{\boldsymbol{S}_{\text{src}}, \boldsymbol{S}_{\text{trg}}\}$, where $\boldsymbol{S}_{\text{src}}, \boldsymbol{S}_{\text{trg}}$ denote the source, target speech, respectively. Our goal is to find a set of prior model parameters to train the final encoder $\text{Enc}^{\text{S}}_{\text{VC}}$ and decoder $\text{Dec}^{\text{S}}_{\text{VC}}$.

4.2.1 TTS-oriented pretraining

In this subsection we review the TTS-oriented pretraining technique [41]. We assume that access to a large single-speaker TTS corpus $\boldsymbol{D}_{\text{TTS}} = \{\boldsymbol{T}_{\text{TTS}}, \boldsymbol{S}_{\text{TTS}}\}$ is available, where $\boldsymbol{T}_{\text{TTS}}, \boldsymbol{S}_{\text{TTS}}$ denote the text and speech of the TTS speaker respectively. The pretraining can be broken down into two steps.

A.1 *Decoder pretraining*: As in A.1 in Figure 4.3, the decoder is pretrained, on $\boldsymbol{D}_{\text{TTS}}$, by training a conventional TTS model composed of a text encoder $\text{Enc}^{\text{T}}_{\text{TTS}}$ and a speech decoder $\mathrm{Dec}^{\mathrm{S}}_{\mathrm{TTS}}$.

A.2 *Encoder pretraining*: Then, as in A.2 in Figure 4.3, the encoder is pretrained, also on the same $\boldsymbol{D}_{\mathrm{TTS}}$, by training an autoencoder which takes $\boldsymbol{S}_{\mathrm{TTS}}$ as input and output. The decoder here is the pretrained $\mathrm{Dec}^{\mathrm{S}}_{\mathrm{TTS}}$ and we fix the parameters so that they are not updated during training. The desired pretrained encoder $\mathrm{Enc}^{\mathrm{S}}_{\mathrm{TTS}}$ can then be obtained by minimizing the reconstruction loss.

The intuition of the encoder pretraining is to obtain an encoder capable of encoding acoustic features into hidden representations that are recognizable by the well pretrained decoder. Another interpretation is that the final pretrained encoder $\mathrm{Enc}^{\mathrm{S}}_{\mathrm{TTS}}$ tries to mimic the text encoder $\mathrm{Enc}^{\mathrm{T}}_{\mathrm{TTS}}$. In the first decoder pretraining step, since text itself contains pure linguistic information, the text encoder $\mathrm{Enc}^{\mathrm{T}}_{\mathrm{TTS}}$ is ensured to learn to encode an effective hidden representation that can be consumed by the decoder $\mathrm{Dec}^{\mathrm{S}}_{\mathrm{TTS}}$. Fixing the decoder in the encoder pretraining process, as a consequence, guarantees the encoder to behave similarly to the text encoder, which is to extract fine-grained, linguistic-information-rich representations.

4.2.2 ASR-oriented pretraining

In this subsection we describe how to extend the TTS-oriented pretraining technique in 4.2.1 to ASR. We assume that a large *multi-speaker* ASR corpus $\boldsymbol{D}_{\mathrm{ASR}} = \{\boldsymbol{S}_{\mathrm{ASR}}, \boldsymbol{T}_{\mathrm{ASR}}\}$ is available, where $\boldsymbol{S}_{\mathrm{ASR}}, \boldsymbol{T}_{\mathrm{ASR}}$ denote the speech and text data in $\boldsymbol{D}_{\mathrm{ASR}}$, respectively. Similar to TTS-oriented pretraining, the ASR-oriented pretraining is again broken down into two steps.

B.1 *Encoder pretraining*: First, the *encoder* is pretrained, on $\boldsymbol{D}_{\mathrm{ASR}}$, by training a conventional ASR model consisting a speech encoder $\mathrm{Enc}^{\mathrm{S}}_{\mathrm{ASR}}$ and a text decoder

$\text{Dec}^{\text{T}}_{\text{ASR}}$, as in B.1 in Figure 4.3.

B.2 *Decoder pretraining*: Differently, the decoder pretraining is performed on $\boldsymbol{D}_{\text{TTS}}$, rather on $\boldsymbol{D}_{\text{ASR}}$. This is because $\boldsymbol{D}_{\text{ASR}}$ is a multi-speaker corpus, but the VC model architecture in this work focuses on one-to-one VC, i.e. modeling the conversion between one source speaker and one target speaker, thus cannot model individual speaker characteristics. Again, the decoder pretraining uses $\boldsymbol{S}_{\text{TTS}}$ as input and output, and the encoder is the pretrained $\text{Enc}^{\text{S}}_{\text{ASR}}$ and kept fixed during training. To speed up convergence, we initialize the decoder with the one obtained in TTS decoder pretraining, namely $\text{Dec}^{\text{S}}_{\text{TTS}}$. The desired pretrained decoder $\text{Enc}^{\text{S}}_{\text{ASR}}$ can then be obtained by minimizing the reconstruction loss. The decoder pretraining procedure is depicted in B.2 in Figure 4.3.

The intuition of the ASR decoder pretraining is different from that of the TTS encoder pretraining. The ASR speech encoder $\text{Enc}^{\text{S}}_{\text{ASR}}$, trained with the ASR objective, should generate a compact hidden representation for decoding underlying linguistic contents. Such representations are believed to be easier to map to speech, thus suitable for pretraining the speech decoder $\text{Dec}^{\text{S}}_{\text{ASR}}$.

4.2.3 VC model training

Finally, as in [41], $\boldsymbol{D}_{\text{VC}}$ is used to train the desired VC models $\text{Enc}^{\text{S}}_{\text{VC}}$ and $\text{Dec}^{\text{S}}_{\text{VC}}$, with the encoder initialized with either $\text{Enc}^{\text{S}}_{\text{TTS}}$ or $\text{Enc}^{\text{S}}_{\text{ASR}}$, and the decoder with $\text{Dec}^{\text{S}}_{\text{TTS}}$ or $\text{Dec}^{\text{S}}_{\text{ASR}}$, respectively. As we will show later, the pretrained model parameters serve as a very good prior to adapt to the relatively scarce VC data, achieving significantly better conversion performance.

4.3 Experimental evaluations

4.3.1 Experimental settings

Data

For $\boldsymbol{D}_{\mathrm{VC}}$, we conducted our experiments on the CMU ARCTIC database [121], which contains parallel recordings of professional US English speakers sampled at 16 kHz. Data from four speakers were used: a male source speaker (*bdl*) and a female source speaker (*clb*), as well as a male target speaker (*rms*) and a female target speaker (*slt*). 100 utterances were selected for each validation and evaluation sets, and the remaining 932 utterances were used as training data. For $\boldsymbol{D}_{\mathrm{TTS}}$, we chose a US female English speaker (*judy bieber*) from the M-AILABS speech dataset [84]. With the sampling rate also at 16 kHz, the training set contained 15,200 utterances, which were roughly 32 hours long. For $\boldsymbol{D}_{\mathrm{ASR}}$, we used the LibriSpeech dataset [81] and pooled *train-clean-100* and *train-clean-360* together to get 460 hours of data from roughly 1170 speakers.

Implementation

The entire experiment was carried out on the open-source ESPnet toolkit [71, 78], including feature extraction, training and benchmarking. The official implementation has been made publicly available[1], and since readers may access all the settings and configurations online, we omit the detailed hyperparameters here. For the VC training, 80-dimensional mel filterbanks with 1024 FFT points and a 256 point frame shift was used as the acoustic features. We used the LAMB optimizer [122] and set the learning rate to 0.001.

[1] `https://github.com/espnet/espnet/tree/master/egs/arctic/vc1`

Waveform synthesis module

We used the Parallel WaveGAN (PWG) [80], which is a non-autoregressive variant of the WaveNet vocoder [27, 28] and enables parallel, faster than real-time waveform generation[2]. Since speaker-dependent neural vocoders outperform speaker-independent ones [29], we trained a speaker-dependent PWG conditioned on natural mel spectrograms, one for each target speaker. Note that we used the full training dataset, since the goal is to demonstrate the effects of various methods, so we did not train separate PWGs w.r.t. different training data sizes.

Objective evaluation metrics

We carried out three types of objective evaluations between the converted speech and the ground truth.

- Mel cepstrum distortion (MCD) [123]: The MCD is a commonly used measure of spectral distortion in VC, which is based on mel-cepstral coefficients (MCCs). It is defined as:

  $$\text{MCD}[dB] = \frac{10}{\log 10}\sqrt{2\sum_{d=1}^{K}(mcc_d^{(c)} - mcc_d^{(t)})^2}, \tag{4.1}$$

  where K is the dimension of the MCCs and $mcc_d^{(c)}$ and $mcc_d^{(t)}$ represent the d-th dimensional coefficient of the converted MCCs and the target MCCs, respectively. In practice, MCD is calculate in a utterance-wise manner. A dynamic time warping (DTW) based alignment is performed to find the corresponding frame pairs between the non-silent converted and target MCC sequences beforehand. We

[2] We followed the open-source implementation at `https://github.com/kan-bayashi/ParallelWaveGAN`

used the WORLD vocoder [12] for MCC extraction and silence frame decisions, and set $K = 24$.

- F0 root mean square error (F0RMSE): Since using seq2seq modeling in VC can greatly improve prosody conversion, we report the RMSE between the F0 of converted speech and that of the reference target speech. Similar to the calculation of MCD, DTW-based is performed and we take only the non-silent frames into account.

- Character/word error rate (CER/WER): The CER/WER is an underestimate of the intelligibility of the converted speech. The ASR engine is based on the Transformer architecture [69] and is trained using the LibriSpeech dataset [81]. The CER and WER for the ground-truth validation set were 0.9% and 3.8%, respectively, which could be regarded as the upper bound.

Note that to avoid overfitting, we used the validation set MCD as the criterion for model selection, and the best performing models were proceeded to generate the samples for the subjective test.

Subjective evaluation methods

The following subjective evaluations were performed using the open-source toolkit [124] which implements the ITU-T Recommendation P.808 [125] for subjective speech quality assessment in the crowd using the Amazon Mechanical Turk (Mturk), and screens the obtained data for unreliable ratings. We recruited more than fifty listeners.[3]

- The mean opinion score (MOS) test on naturalness: Subjects were asked to

[3] A demo web page with samples used for subjective evaluation is available at `https://unilight.github.io/Publication-Demos/publications/vtn-taslp/index.html`

evaluate the naturalness of the converted and natural speech samples on a scale from 1 (completely unnatural) to 5 (completely natural).

- The VCC [43] style test on similarity: This paradigm was adopted by the VCC organizing committee. Listeners were given a pair of speech utterances consisting of a natural speech sample from a target speaker and a converted speech sample. Then, they were asked to determine whether the pair of utterances can be produced by the same speaker, with 4-level confidence of their decision, i.e., sure or not sure.

4.3.2 Effectiveness of TTS-oriented pretraining on RNN and Transformer based models

First, we show that TTS-oriented pretraining is a technique effective on not only VTN but also RNN-based seq2seq VC models. The objective results are in Table 4.1. First, without pretraining, both VTN and RNN could not stay robust against the reduction of training data. The performance dropped dramatically with the reduction of training data, where a similar trend was also reported in [58]. This identifies the data efficiency problem of seq2seq VC. By incorporating TTS-oriented pretraining, both VTN and RNN exhibited a significant improvement in all objective measures, where the effectiveness was robust against the reduction in the size of training data. With only 80 utterances, both models can achieve comparable performance to that of using the full training dataset except the F0RMSE, wherein the case of VTN, the intelligibility is even better.

The subjective results are in Table 4.2. Without pretraining, the VTN and RNN suffered from about 1.2 and 0.8 MOS points drop when the training data reduces from

932 to 80 utterances. On the other hand, with TTS-oriented pretraining applied, the naturalness of VTN and RNN improved by more than 1 point with 932 utterances and more than 2 points with 80 utterances. Moreover, when the training data reduces, there was only a very limited performance drop. These results demonstrate the effectiveness of the TTS-oriented pretraining technique.

4.3.3 Comparison of TTS-oriented and ASR-oriented pretraining

Next, we compare the effectiveness of TTS-oriented and ASR-oriented pretraining. From Tables 4.1 and 4.2, with the full training set, ASR-oriented pretraining could bring almost the same amount of improvement compared to TTS-oriented pretraining. However, as the size of training data reduces, the performance of the ASR-oriented pretrained model dropped significantly, except F0RMSE. This shows that ASR-oriented pretraining lacks the robustness essential for practical VC.

To investigate the failure of ASR-oriented pretrained models against limited training data, we chose one sentence from the evaluation set and show the ASR results of the converted samples using TTS-oriented and ASR-oriented pretrained VTNs in Table 4.3. Although TTS-oriented pretraining could not ensure complete linguistic consistency, the errors were minor and possibly due to the imperfect ASR engine used for evaluation, thus the result seems reasonable. On the other hand, the recognition result of the ASR-oriented pretrained model with 80 utterances that had no connection to the source sentence. We conclude that linguistic consistency is poorly maintained under the limited data scenario using ASR-oriented pretraining.

4.3.4 Comparison of RNN and Transformer based models

In [41], the VTN was shown to outperform the RNN-based model [39], while it was not clear whether the improvement came from different model architectures or the pretraining technique. To make a clearer comparison, we applied TTS-oriented pretraining to both VTNs and RNNs. From Tables 4.1 and 4.2, it was shown that without TTS-oriented pretraining, VTNs were less robust to training data reduction than RNNs in terms of objective measures but better in terms of subjective measures. This is possibly because that a more complex model like VTN is capable of generating better-sounding voices while being more prone to overfitting since it lacks attention regularizations such as the location-sensitive location, as suggested in [71]. As we applied TTS-oriented pretraining to both VTN and RNN, it could be clearly observed that VTNs outperformed RNNs in terms of all objective measures except F0RMSE and subjective scores. This is possibly due to the use of MCD as the model selection criterion.

4.3.5 Visualizing the hidden representation space

In Section 4.2.1, we suspected that applying the TTS-oriented pretraining technique results in an encoder that can extract linguistic-information-rich representation. To demonstrate this tendency, we extracted the hidden representations with the trained encoders using the validation set from the *clb* speaker as input, and visualized them using the t-SNE method [126]. We used the phoneme labels that come with the CMU ARCTIC dataset as ground truth and colored the 5 most common phonemes and their corresponding hidden representations to simplify the plots. Note that for encoders with a reduction factor greater than 1, the corresponding label was decided with majority voting. For example, if the encoder reduction factor is 4, and the labels of the four frames that corresponds to a hidden representation are ”s”, ”s”, ”s”, ”a”, then the label of that hidden representation will be set to ”s”.

The resulting plots are shown in Figure 4.4. It could be clearly observed that compared to no pretraining, the hidden representation spaces learned from TTS-oriented pretraining demonstrated a strong degree of clustering effect, where points correspond to the same phoneme were close to each other. This tendency was consistent in the cases of both 932 and 80 training utterances. On the other hand, ASR-oriented pretraining yielded a much scatter hidden representation space even with 932 training utterances.

This analysis suggests that the TTS-oriented pretraining technique can result in a more discretized representation space, which matches our initial assumption. We may further conclude that, by looking together with the objective and subjective results in Tables 4.1 and 4.2, the degree of clustering effect somehow reflect the goodness of the hidden representations for seq2seq VC.

4.4 Conclusions

In this work, we evaluated the pretraining techniques for addressing the problem of data efficiency in seq2seq VC. Specifically, a unified, two-stage training strategy that first pretrains the decoder and the encoder subsequently followed by initializing the VC model with the pretrained model parameters was proposed. ASR and TTS were chosen as source tasks to transfer knowledge from, and the RNN and VTN architectures were implemented. Through objective and subjective evaluations, it was shown that the TTS-oriented pretraining strategy can greatly improve the performance in terms of speech intelligibility and quality when applied to both RNNs and VTNs, and the performance could stay without significant degradation even with limited training data. As for ASR-oriented pretraining, the robustness was not so good with the reduction of training data size. Also, VTNs performed inferior to RNNs without pretraining but superior with TTS-oriented pretraining. The visualization experiment suggested that the TTS-oriented pretraining could learn a linguistic-information-rich hidden representation space while the ASR-oriented pretraining lacks such ability, which lets us imagine what an ideal hidden representation space would be like.

In the future, we plan to extend our pretraining technique to more flexible training conditions, such as many-to-many [127] or nonparallel training [58].

Table 4.1: *Validation-set objective evaluation results of VTNs with no pretraining, TTS-oriented pretraining, ASR-oriented pretraining, and RNN-based models with no pretraining and TTS-oriented pretraining, which are trained on different conversion pairs and different sizes of data. Bold font indicates the best performance across the average scores.*

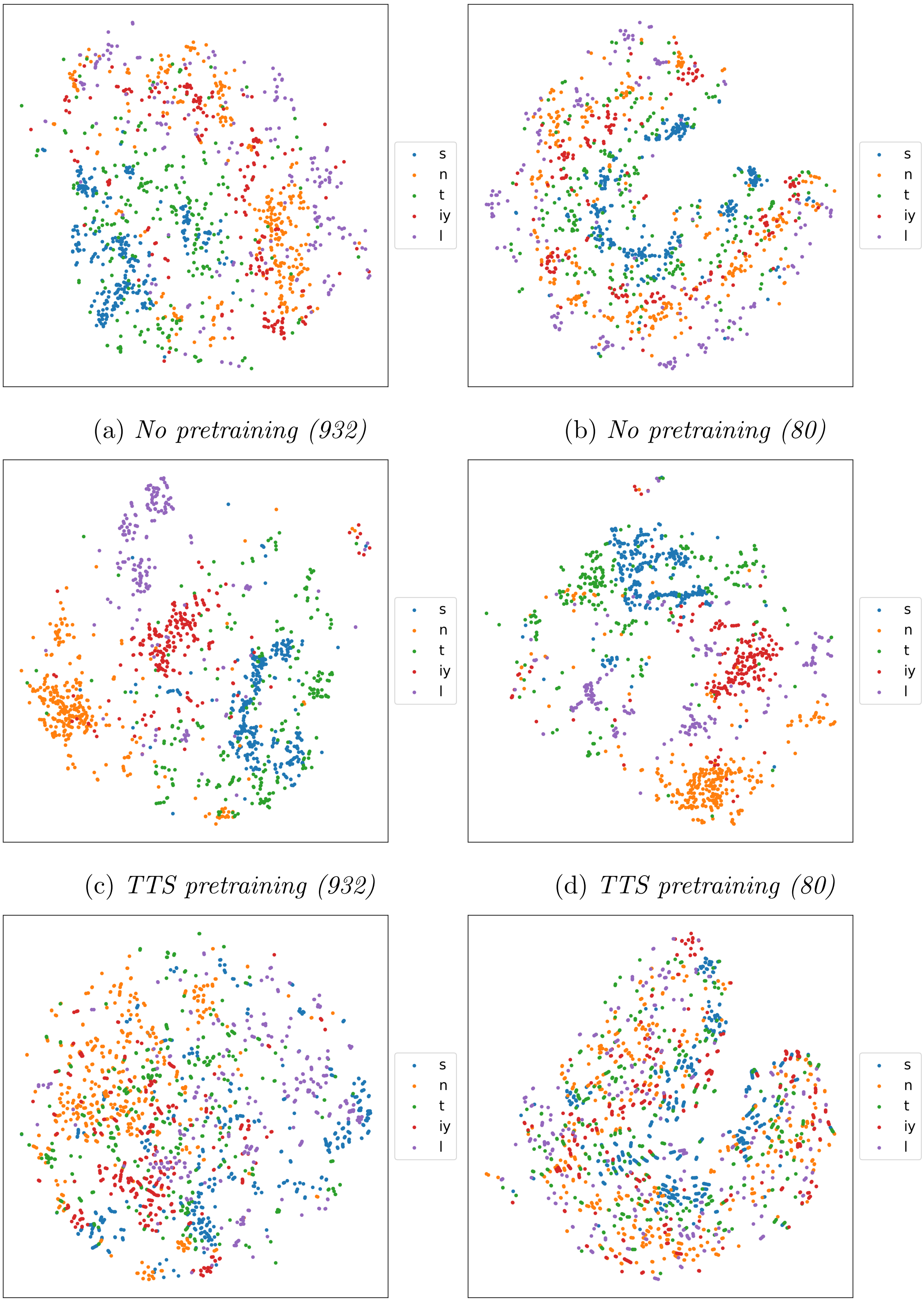
		Pair		932 training utterances				250 training utterances				80 training utterances			
Model	Pretrain	Src	Trg	MCD	F0RMSE	CER	WER	MCD	F0RMSE	CER	WER	MCD	F0RMSE	CER	WER
VTN	None	clb	slt	6.60	24.54	12.4	20.2	7.43	25.37	29.2	42.3	8.23	26.65	65.3	87.6
			rms	6.83	24.02	21.4	33.0	7.83	24.86	53.2	73.3	8.68	27.20	71.8	94.8
		bdl	slt	7.33	23.36	23.1	33.7	8.31	23.40	52.9	75.9	8.74	24.81	73.9	95.7
			rms	7.37	22.68	28.4	43.3	8.30	23.78	56.2	78.4	9.14	24.38	79.0	102.8
		Average		7.03	23.65	21.3	32.6	7.97	24.35	47.9	67.5	8.70	25.76	72.5	95.2
	TTS	clb	slt	6.02	23.94	5.5	9.1	6.41	24.86	5.2	9.7	6.66	27.24	10.4	14.7
			rms	6.22	24.29	6.8	11.9	6.75	24.91	12.8	21.3	6.94	27.86	12.5	22.0
		bdl	slt	6.33	22.44	5.0	7.6	6.71	23.24	4.8	8.1	7.07	23.75	9.7	13.6
			rms	6.69	22.63	7.3	12.7	7.13	23.32	11.3	18.0	7.39	24.13	17.2	26.2
		Average		**6.32**	23.33	**6.2**	**10.3**	**6.75**	**24.08**	**8.5**	**14.3**	**7.02**	25.75	**12.5**	**19.1**
	ASR	clb	slt	6.11	24.03	4.8	10.9	6.84	24.78	15.9	26.0	8.28	27.13	72.1	97.6
			rms	6.22	24.15	8.1	16.0	7.08	24.89	27.2	43.2	7.93	26.57	60.2	86.2
		bdl	slt	6.50	22.35	5.7	11.1	7.33	23.65	26.1	39.8	8.18	24.23	58.2	80.7
			rms	6.68	22.46	9.1	15.6	7.58	23.24	32.9	51.6	8.22	24.25	59.7	82.9
		Average		6.38	**23.25**	6.9	13.4	7.21	24.14	25.5	40.2	8.15	25.55	62.6	86.9
RNN	None	clb	slt	6.77	24.81	7.1	12.1	7.29	25.02	15.4	24.0	7.76	25.04	38.6	56.8
			rms	6.80	23.54	11.6	19.7	7.49	24.84	24.7	38.0	7.98	27.67	48.9	68.7
		bdl	slt	7.45	23.37	23.4	32.6	8.06	24.53	37.1	54.4	8.44	24.40	65.6	93.8
			rms	7.62	23.96	20.0	32.4	8.25	24.32	47.2	90.2	8.52	25.13	59.7	81.5
		Average		7.16	23.92	15.5	24.2	7.77	24.68	31.1	51.7	8.18	25.56	53.2	75.2
	TTS	clb	slt	6.29	24.62	5.6	10.1	6.63	23.99	7.4	12.7	6.92	26.40	14.0	22.2
			rms	6.35	23.58	8.3	16.1	6.88	24.30	17.0	27.7	7.08	26.54	29.0	44.0
		bdl	slt	6.74	22.89	8.2	13.9	7.08	23.11	11.3	19.8	7.46	23.60	16.3	23.8
			rms	6.97	22.36	15.1	26.3	7.39	23.34	21.1	32.4	7.57	23.30	25.4	39.6
		Average		6.59	23.36	9.3	16.6	7.00	23.69	14.2	23.2	7.26	**24.96**	21.2	32.4

Table 4.2: *Evaluation-set naturalness and similarity subjective evaluation results of VTNs with no pretraining, TTS pretraining, ASR pretraining, and RNN-based models with no pretraining and TTS pretraining, which are averaged over all conversion pairs and different sizes of data.*

		932 training utterances		80 training utterances	
Model	Pretraining	Naturalness	Similarity	Naturalness	Similarity
Analysis-synthesis		4.45 ± 0.14	-	-	-
VTN	None	3.19 ± 0.23	61% ± 14%	1.96 ± 0.16	44% ± 13%
	TTS	4.34 ± 0.15	80% ± 11%	4.11 ± 0.09	68% ± 8%
	ASR	4.25 ± 0.16	77% ± 10%	3.38 ± 0.20	53% ± 12%
RNN	None	2.33 ± 0.20	40% ± 12%	1.57 ± 0.14	33% ± 15%
	TTS	3.91 ± 0.19	68% ± 13%	3.71 ± 0.09	58% ± 10%

Table 4.3: *ASR-based recognition results of VTN converted samples from the evaluation set of the clb-slt conversion pair. The errors are in uppercase.*

Description	Training data size	Recognition result
Ground truth	-	the history of the eighteenth century is written ernest prompted
TTS-oriented pretraining	932	the history of the eighteenth century is written IN IS prompted TO TO
	250	the history of the eighteenth century is written IN HIS PROMPTER
	80	the history of the eighteenth century is written ON HIS PROMPT
ASR-oriented pretraining	932	the history of the eighteenth century is written EARNEST prompted
	250	the history of the eighteenth CENTURY'S RADIANCE prompted
	80	IT DISTURBED the DAY TO HIMSELF TO REJOIN HIM IN NORTH'S LIBRARY



(a) *No pretraining (932)* (b) *No pretraining (80)*

(c) *TTS pretraining (932)* (d) *TTS pretraining (80)*

(e) *ASR pretraining (932)* (f) *ASR pretraining (80)*

Figure 4.4: *Visualizations of hidden representations extracted from VTNs with no pretraining, TTS-oriented pretraining, and ASR-oriented pretraining. The validation set from clb was used. The numbers in the parenthesis indicate the number of training utterances.*

5 Conclusions

5.1 Summary of the Thesis

In this thesis, we studied the problem of VC, a technique that can transform the characteristics of the source speech into that of the target speech such that the linguistic contents are preserved. VC can be beneficial in various real-world scenarios, including entertainment, education and medical applications. We focused on developing seq2seq VC models, which is superior to conventional frame-wise models in terms of the modeling of prosody. To overcome the data deficiency problem, two transfer learning approaches have been presented. It is believed that such techniques can make seq2seq VC models more practical to use.

Chapter 2 described the seq2seq VC model architectures used in this thesis. We first observed that TTS and VC share a common objective of synthesizing speech waveform, and thus suggested a seq2seq VC model can be easily obtained by making simple modifications to a TTS model. We then give a unified formulation of a seq2seq model and some common components for the general speech synthesis task. Finaly, we described two seq2seq VC models in detail, nemaly the RNN-based model and the Transformer-based model.

In Chapter 3, a system that consists a cascade of two readily trained seq2seq ASR and TTS model was presented. We argued that simply concatenating state-of-the-art ASR and TTS models that are trained with large-scale datasets in the respective research fields can yield superior performance even with limited training data. The proposed system was adopted to be the seq2seq baseline of the VCC2020, and through the official listening test, it was shown that the system served as a competitive baseline system, even ranking second in terms of speaker siilarity. This result demonstrated the effectiveness of the proposed framework. In addition, the system has been made publicly available for the participants, and will maintain open source to benefit future potential interested researchers.

In Chapter 4, we focused on the optimization of one unified seq2seq VC model in a parallel, one-to-one setting. By observing VC, TTS and ASR in an information perspective, we proposed to transfer the most essential ability of a seq2seq model, i.e, the extraction of meaningful hidden representations, from ASR and TTS. The proposed two-step pretraining-finetuning scheme serves a prior for fast, sample-efficient seq2seq VC model learning. Through objective and subjective evaluations, we demonstrated that both ASR-oriented and TTS-oriented pretraining could improve the performance with adequate data, but only TTS-oriented pretraining maintained robust with limited data. We also showed the superiority of Transformer-based models over RNN-based ones.

5.2 Future Work

The goal of developing transfer leaning techniques to mitigate the data deficiency problem is to make seq2seq VC models more applicable to real-world scenarios. However, there are still a number of research questions that needs to be solved.

5.2.1 Real-time, low-latency processing

Despite the promising accuracy, seq2seq VC models are computationally expensive and memory consuming, thus needs optimization. In VC, the latency criterion for production level VC is 50 ms, but in the current seq2seq VC, the latency is 600 ms. This is due to the massive parameters of DNNs and the autoregressive decoding process of seq2seq. In the fields of ASR and TTS, several techniques have been explored to build seq2seq models in mobile devices [128,129]. To build VC devices for patients and users, such investigation is necessary.

Specifically, to tackle these problems, general methods for compressing DNNs including parameter quantization and weight pruning [130] and methods specialized for seq2seq models such as streaming decoding with monotonic chunkwise attention [131] can be applied. It is expected that by evaluating methods for compressing and accelerating the seq2seq VC model, the computational and latency constraints can be satisfied by applying valid techniques.

5.2.2 Augmented speech communication with multi-modal signals

In addition to using speech organs to produce speech, we seek an interactive speech production paradigm by utilizing what we refer to as a multimodal signal, depicted in Figure 5.1. Such signals can be collected from human body movements or devices such as digital instruments, thus can be viewed as an extension of human intention. By modeling the multimodal signals, we may more flexibly control various aspects of speech, such as timbre, emotion, accent, etc. For instance, in addition to developing speaking aid devices for patients with speech organ disabled, we may further integrate

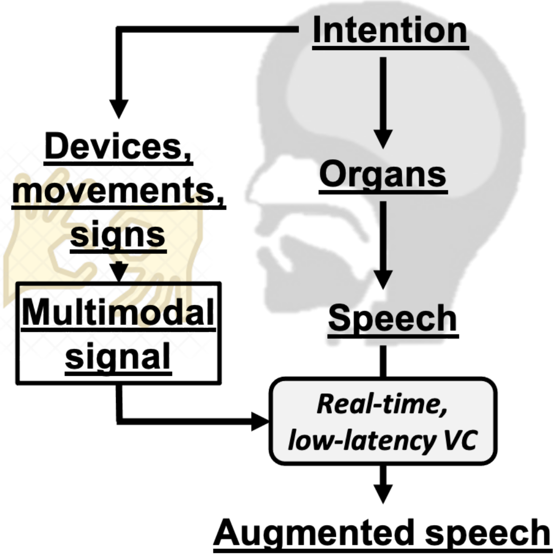


Figure 5.1: *Augmented speech communication.*

musical instruments [132, 133] or visual sensors capturing the facial expression so that users may produce speech with the intended emotion, making it possible for patients to more efficiently convey non-linguistic information. In a nutshell, a paradigm shift that relies on visual or behavior signals like signs and movements is believed to drastically change the form of human-to-human communication beyond physical barriers.

Technically speaking, a new multi-modal dataset needs to be collected first. The plan is to collect several datasets, including (1) the singing voices and the signals from digital keyboard players, (2) emotional speech of voice actors with facial expression and hand gestures signals. Once such a dataset is collected, the previously proposed seq2seq VC model can be deployed to test the feasibility of using the multi-modal signal. As it is impractical to collect a large amount of such data, the pretraining technique should be essential to alleviate data scarcity. The ultimate goal is to combine the real-time, low-latency techniques described in Subsection 5.2.1 to form a uniform system for the prototype demo. The datasets and the developed systems can be freely available for further research purposes.

Acknowledgments

I would like to convey my deepest gratitude to my thesis advisor, Professor Tomoki Toda of Nagoya University, for everything he has done to support me throughout my Master's study. He has always been my role model as a researcher, and his expertise and insightful feedback have always pushed me to sharpen my thinking and brought my work to a higher level.

I would like to acknowledge Professor Hsin-Min Wang of the Institute of Information Science, Academia Sinica Taipei and Professor Yu Tsao of the Research Center for Information Technology Innovation, Academia Sinica Taipei, for their patient support and for all of the opportunities I was given to further my research.

I would like to thank Associate Professor Hirokazu Kameoka of NTT Corporation for the invaluable opportunity to intern at the NTT Communication Science Laboratory.

I would especially like to express my humble gratefulness to NTT Docomo for the financial support of my research work.

I would like to devote my earnest thanks to the staff of Toda Laboratory for their kind assistance. I would also like to convey my thanks to laboratory colleagues for their support, especially Dr. Kazuhiro Kobayashi of TARVO, Inc., Dr. Tomoki Hayashi of Human Dataware Lab. Co., Ltd., and Mr. Yi-Chiao Wu, for their aids and advice on my research.

I would also like to thank my colleagues from the Speech, Language, and Music

Processing Lab when I worked as a research assistant in the Institute of Information Science, Academia Sinica Taipei, for the fruitful, inspiring discussion we had and the great memories we shared. I would particularly like to single out Dr. Hsin-Te Hwang, for teaching me the fundamentals of conducting scientific research.

Finally, I would like to express my wholehearted recognition to my family for their warm company and wise counsel. I would also like to thank the many friends I met in Japan. You enriched my life outside of my research.

References

[1] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[2] D. Childers, B. Yegnanarayana, and K. Wu, "Voice conversion: Factors responsible for quality," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 10, 1985, pp. 748–751.

[3] S. Aryal, D. Felps, and R. Gutierrez-Osuna, "Foreign accent conversion through voice morphing." in *Proc. Interspeech*, 2013, pp. 3077–3081.

[4] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1998, pp. 285–288.

[5] J. Latorre, V. Wan, and K. Yanagisawa, "Voice expression conversion with factorised hmm-tts models," in *Proc. Interspeech*, 2014, pp. 1514–1518.

[6] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, 2012.

[7] T. Toda, K. Nakamura, H. Saruwatari, K. Shikano *et al.*, "Alaryngeal speech enhancement based on one-to-many eigenvoice conversion," *IEEE/ACM Trans-*

actions on Audio, Speech and Language Processing (TASLP), vol. 22, no. 1, pp. 172–183, 2014.

[8] K. Tanaka, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, “A hybrid approach to electrolaryngeal speech enhancement based on spectral subtraction and statistical voice conversion.” in *Proc. Interspeech*, 2013, pp. 3067–3071.

[9] T. Toda, “Augmented speech production based on real-time statistical voice conversion,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 592–596.

[10] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, “Foreign accent conversion in computer assisted pronunciation training,” *Speech communication*, vol. 51, no. 10, pp. 920–932, 2009.

[11] T. Toda, A. W. Black, and K. Tokuda, “Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[12] M. Morise, F. Yokomori, and K. Ozawa, “WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications,” *IEICE Transactions on Information and Systems*, vol. 99, pp. 1877–1884, 2016.

[13] H. Kawahara, I. Masuda-Katsuse, and A. de CheveignÃ©, “Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds,” *Speech Communication*, vol. 27, no. 3, pp. 187–207, 1999.

[14] Y. Stylianou, O. Cappe, and E. Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[15] D. Sundermann and H. Ney, “Vtln-based voice conversion,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003, pp. 556–559.

[16] D. Erro, A. Moreno, and A. Bonafonte, “Voice conversion based on weighted frequency warping,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 18, no. 5, pp. 922–931, 2009.

[17] E. Godoy, O. Rosec, and T. Chonavel, “Voice conversion using dynamic frequency warping with amplitude scaling, for parallel or nonparallel corpora,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 20, no. 4, pp. 1313–1323, 2011.

[18] R. Takashima, T. Takiguchi, and Y. Ariki, “Exemplar-based voice conversion in noisy environment,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 313–317.

[19] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, “Exemplar-based sparse representation with residual compensation for voice conversion,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1506–1521, 2014.

[20] Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, “Locally linear embedding for exemplar-based spectral conversion,” in *Proc. Interspeech*, 2016, pp. 1652–1656.

[21] B. Sisman, H. Li, and K. C. Tan, "Sparse representation of phonetic features for voice conversion with and without parallel data," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 677–684.

[22] X. Tian, S. W. Lee, Z. Wu, E. S. Chng, and H. Li, "An exemplar-based approach to frequency warping for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 25, no. 10, pp. 1863–1876, 2017.

[23] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana, "Transformation of formants for voice conversion using artificial neural networks," *Speech communication*, vol. 16, no. 2, pp. 207–216, 1995.

[24] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[25] L. H. Chen, Z. H. Ling, L. J. Liu, and L. R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[26] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4869–4873.

[27] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[28] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent wavenet vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.

[29] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for WaveNet vocoder," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 712–718.

[30] K. Kobayashi, T. Hayashi, A. Tamamori, and T. Toda, "Statistical voice conversion with wavenet-based waveform generation," in *Proc. Interspeech*, 2017, pp. 1138–1142.

[31] B. Sisman, M. Zhang, and H. Li, "A voice conversion framework with tandem feature sparse representation and speaker-adapted "wavenet" vocoder," in *Proc. Interspeech*, 2018, pp. 1978–1982.

[32] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "Wavenet vocoder with limited training data for voice conversion," in *Proc. Interspeech*, 2018, pp. 1983–1987.

[33] P. L. Tobing, T. Hayashi, Y.-C. Wu, K. Kobayashi, and T. Toda, "An evaluation of deep spectral mappings and wavenet vocoder for voice conversion," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 297–303.

[34] W.-C. Huang, Y.-C. Wu, H.-T. Hwang, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, Y. Tsao, and H.-M. Wang, "Refined wavenet vocoder for variational autoencoder based voice conversion," in *27th European Signal Processing Conference (EUSIPCO)*, 2019.

[35] B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, "Adaptive wavenet vocoder for residual compensation in gan-based voice conversion," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 282–289.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[37] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[38] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, Sep. 2015, pp. 1412–1421.

[39] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "ATTS2S-VC: Sequence-to-sequence Voice Conversion with Attention and Context Preservation Mechanisms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 6805–6809.

[40] J. Zhang, Z. Ling, L. Liu, Y. Jiang, and L. Dai, "Sequence-to-Sequence Acoustic Modeling for Voice Conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 631–644, 2019.

[41] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," *arXiv preprint arXiv:1912.06813*, 2019, to appear in Interspeech 2020.

[42] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, “The voice conversion challenge 2016,” in *Proc. Interspeech*, 2016, pp. 1632–1636.

[43] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, “The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods,” in *Proc. Odyssey*, 2018, pp. 195–202.

[44] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Toda, T. Kinnunen, and Z. Ling, “Voice conversion challenge 2020 — intra-lingual semiparallel and cross-lingual voice conversion —,” in *ISCA Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020.

[45] J.-X. Zhang, Z.-H. Ling, Y. Jiang, L.-J. Liu, C. Liang, and L.-R. Dai, “Improving Sequence-to-sequence Voice Conversion by Adding Text-supervision,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6785–6789, 2019.

[46] B. Sisman, J. Yamagishi, S. King, and H. Li, “An overview of voice conversion and its challenges: From statistical modeling to deep learning,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132–157, 2021.

[47] J. Wu, Z. Wu, and L. Xie, “On the use of I-vectors and average voice model for voice conversion without parallel data,” in *Proc. APSIPA*, 2016, pp. 1–6.

[48] X. Tian, E. S. Chng, and H. Li, “A Speaker-Dependent WaveNet for Voice Conversion with Non-Parallel Data,” in *Proc. Interspeech*, 2019, pp. 201–205.

[49] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual Voice Conversion with Bilingual Phonetic Posteriorgram and Average Modeling," in *Proc. ICASSP*, 2019, pp. 6790–6794.

[50] S. Liu, Y. Cao, and H. Meng, "Multi-target emotional voice conversion with neural vocoders," *arXiv preprint arXiv:2004.03782*, 2020.

[51] M. Zhang, B. Sisman, S. S. Rallabandi, H. Li, and L. Zhao, "Error reduction network for dblstm-based voice conversion," in *Proc. APSIPA*, 2018, pp. 823–828.

[52] M. Zhang, X. Wang, F. Fang, H. Li, and J. Yamagishi, "Joint Training Framework for Text-to-Speech and Voice Conversion Using Multi-Source Tacotron and WaveNet," in *Proc. Interspeech*, 2019, pp. 1298–1302.

[53] H.-T. Luong and J. Yamagishi, "Nautilus: a versatile voice cloning system," *arXiv preprint arXiv:2005.11004*, 2020.

[54] H. Luong and J. Yamagishi, "Bootstrapping non-parallel voice conversion from speaker-adaptive text-to-speech," in *Proc. ASRU*, 2019, pp. 200–207.

[55] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.

[56] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign Accent Conversion by Synthesizing Speech from Phonetic Posteriorgrams," in *Proc. Interspeech*, 2019, pp. 2843–2847.

[57] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An End-to-End Speech-to-Speech Conversion Model and its Applications to Hearing-

Impaired Speech and Speech Separation," in *Proc. Interspeech*, 2019, pp. 4115–4119.

[58] J. Zhang, Z. Ling, and L. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 540–552, 2020.

[59] H. Kameoka, K. Tanaka, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *CoRR*, vol. abs/1811.01609, 2018.

[60] P. Narayanan, P. Chakravarty, F. Charette, and G. Puskorius, "Hierarchical sequence to sequence voice conversion with limited data," *arXiv preprint arXiv:1907.07769*, 2019.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[62] Y. Wang, R. 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*, 2017, pp. 4006–4010.

[63] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on MEL Spectrogram Pre-

dictions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[64] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[65] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4784–4788.

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

[67] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional Sequence to Sequence Learning," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 1243–1252.

[68] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural Speech Synthesis with Transformer Network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[69] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[70] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs RNN in speech applications," *arXiv preprint arXiv:1909.06317*, 2019.

[71] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, “ESPnet-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text-to-Speech Toolkit,” *arXiv preprint arXiv:1910.10909*, 2019.

[72] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, “WaveNet Vocoder with Limited Training Data for Voice Conversion,” in *Proc. Interspeech*, 2018, pp. 1983–1987.

[73] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, “Voice conversion from non-parallel corpora using variational auto-encoder,” in *Proc. APISPA ASC*, 2016, pp. 1–6.

[74] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, “Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks,” in *Proc. Interspeech*, 2017, pp. 3364–3368.

[75] J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee, “Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations,” in *Proc. Interspeech*, 2018, pp. 501–505.

[76] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, “AutoVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5210–5219.

[77] W.-C. Huang, H. Luo, H.-T. Hwang, C.-C. Lo, Y.-H. Peng, Y. Tsao, and H.-M. Wang, “Unsupervised Representation Disentanglement Using Cross Domain Features and Adversarial Learning in Variational Autoencoder Based Voice Con-

version," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–12, 2020.

[78] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[79] K. Inoue, S. Hara, M. Abe, T. Hayashi, R. Yamamoto, and S. Watanabe, "Semi-supervised speaker adaptation for end-to-end speech synthesis with pretrained models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7634–7638.

[80] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," *arXiv preprint arXiv:1910.11480*, 2019.

[81] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[82] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. Interspeech*, 2019, pp. 1408–1412.

[83] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[84] Munich Artificial Intelligence Laboratories GmbH, "The M-AILABS speech dataset," 2019, accessed 30 November 2019. [Online]. Available: https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/

[85] K. Park and T. Mulc, "CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages," in *Proc. Interspeech*, 2019, pp. 1566–1570.

[86] Data Baker China, "Chinese standard mandarin speech corpus," accessed 05 May 2020. [Online]. Available: {www.data-baker.com/open_source.html}

[87] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," in *Proc. Interspeech*, 2019, pp. 2080–2084.

[88] L. Xue, W. Song, G. Xu, L. Xie, and Z. Wu, "Building a mixed-lingual neural tts system with only monolingual data," in *Proc. Interspeech*, 2019, pp. 2060–2064.

[89] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[90] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

[91] H.-T. Luong, X. Wang, J. Yamagishi, and N. Nishizawa, "Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora," in *Proc. Interspeech*, 2019, pp. 1303–1307.

[92] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Science and Language*, 2019.

[93] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, “Transfer learning from speaker verification to multispeaker text-to-speech synthesis,” in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[94] W.-N. Hsu, Y. Zhang, R. Weiss, H. Zen, Y. Wu, Y. Cao, and Y. Wang, “Hierarchical generative modeling for controllable speech synthesis,” in *International Conference on Learning Representations*, 2019.

[95] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[96] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[97] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition,” in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 1, 2014, pp. 647–655.

[98] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[99] R. Girshick, “”fast r-cnn”,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[100] S. Ren, K. He, R. Girshick, and J. Sun, “”faster r-cnn: Towards real-time object detection with region proposal networks”,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

[101] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[102] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “”mask r-cnn”,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[103] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[104] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European Conference on Computer Vision*, 2016, pp. 694–711.

[105] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Jun. 2018, pp. 2227–2237.

[106] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jul. 2018, pp. 328–339.

[107] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[108] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019, pp. 4171–4186.

[109] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[110] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE TASLP*, vol. 20, no. 1, pp. 30–42, 2011.

[111] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[112] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[113] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Proc. Interspeech*, 2019, pp. 161–165.

[114] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, "VQVAE Unsupervised Unit Discovery and Multi-Scale Code2Spec Inverter for Zerospeech Challenge 2019," in *Proc. Interspeech*, 2019, pp. 1118–1122.

[115] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.

[116] Y.-A. Chung and J. Glass, "Improved speech representations with multi-target autoregressive predictive coding," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2353–2358.

[117] M. Ravanelli and Y. Bengio, "Learning Speaker Representations with Mutual Information," in *Proc. Interspeech*, 2019, pp. 1153–1157.

[118] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[119] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech 2019*, 2019, pp. 3465–3469.

[120] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations," in *International Conference on Learning Representations*, 2020.

[121] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[122] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," in *Proc. ICLR*, 2020.

[123] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, 1993, pp. 125–128 vol.1.

[124] B. Naderi and R. Cutler, "An open source implementation of itu-t recommendation p. 808 with validation," *arXiv preprint arXiv:2005.08138*, 2020.

[125] ITU-T Recommendation P.808, *Subjective evaluation of speech quality with a crowdsourcing approach*, Std., 2018.

[126] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[127] H. Kameoka, W.-C. Huang, K. Tanaka, T. Kaneko, N. Hojo, and T. Toda, "Many-to-many voice transformer network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–1, 2020.

[128] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6381–6385.

[129] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient neural audio synthesis,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 2410–2419.

[130] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.

[131] C.-C. Chiu* and C. Raffel*, “Monotonic chunkwise attention,” in *International Conference on Learning Representations*, 2018.

[132] K. Morikawa and T. Toda, “Electrolaryngeal speech modification towards singing aid system for laryngectomees,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 610–613.

[133] L. Li, T. Toda, K. Morikawa, K. Kobayashi, and S. Makino, “Improving singing aid system for laryngectomees with statistical voice conversion and VAE-SPACE,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 784–790.

List of Publications

Journal Papers

1. W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, T. Toda, “Pretraining techniques for sequence-to-sequence voice conversion,” IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 29, Jan. 2021

2. H. Kameoka, W.-C. Huang, K. Tanaka, T. Kaneko, N. Hojo, T. Toda, “Many-to-many voice transformer network,” IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 29, pp. 656-670, Jan. 2021.

3. X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, “ Asvspoof 2019: a large-scale public database of synthetized, converted and replayed speech, ” Computer Speech and Language, vol. 64, p. 101114, 2020

International Conferences

1. Z. Yi, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z.-H. Ling, and T. Toda, "Voice Conversion Challenge 2020 – Intra-lingual semi-parallel and cross-lingual voice conversion –," in Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020, 2020, pp. 80–98

2. R. K. Das, T. Kinnunen, W.-C. Huang, Z.-H. Ling, J. Yamagishi, Z. Yi, X. Tian, and T. Toda, "Predictions of Subjective Ratings and Spoofing Assessments of Voice Conversion Challenge 2020 Submissions," in Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020, 2020, pp. 99–120

3. W.-C. Huang, T. Hayashi, S. Watanabe, and T. Toda, "The Sequence-to-Sequence Baseline for the Voice Conversion Challenge 2020: Cascading ASR and TTS," Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020, 2020, pp. 160-–164

4. W.-C. Huang, P. L. Tobing, Y.-C. Wu, K. Kobayashi, and T. Toda, "The NU Voice Conversion System for the Voice Conversion Challenge 2020: On the Effectiveness of Sequence-to-sequence Models and Autoregressive Neural Vocoders," Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020, 2020, pp. 165—169

5. W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice Transformer Network: Sequence-to-Sequence Voice Conversion Using Transformer with Text-to-Speech Pretraining," Proc. Interspeech, 2020, pp. 4676–4680

6. W.-C. Huang, Y.-C. Wu, K. Kobayashi, Y.-H. Peng, H.-T. Hwang, P. L. Tobing, T. Toda, Y. Tsao, and H.-M. Wang, "Generalization of Spectrum Differential based

Direct Waveform Modification for Voice Conversion," Proc. ISCA Speech Synthesis Workshop (SSW), 2019, pp. 57-–62

7. W.-C. Huang, Y.-C. Wu, C.-C. Lo, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, Y. Tsao, and H.-M. Wang, "Investigation of F0 Conditioning and Fully Convolutional Networks in Variational Autoencoder Based Voice Conversion,", Proc. Interspeech, 2019, pp. 709-–713

8. C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep Learning based Objective Assessment for Voice Conversion," in Proc. Interspeech, 2019, pp. 1541-–1545

9. W.-C. Huang, Y.-C. Wu, H.-T. Hwang, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, Y. Tsao, H.-M. Wang. "Refined WaveNet vocoder for variational autoencoder based voice conversion," Proc. EUSIPCO, 2019, pp. 1–5.

Awards

1. Scholarship for International Students, JEES Docomo, Apr. 2019 – Mar. 2021

2. Travel grant, ISCA and Interspeech 2019

Academic Activities

1. Organizing Committee, Voice Conversion Challenge 2020