ABSTRACT

Voice conversion (VC) and text-to-speech (TTS) are two tasks that share a similar objective, generating speech with a target voice. However, they are usually developed independently under vastly different frameworks. In this paper, we propose a methodology to bootstrap a VC system from a pretrained speaker-adaptive TTS model and unify the techniques as well as the interpretations of these two tasks. Moreover by offloading the heavy data demand to the training stage of the TTS model, our VC system can be built using a small amount of target speaker speech data. It also opens up the possibility of using speech in a foreign unseen language to build the system. Our subjective evaluations show that the proposed framework is able to not only achieve competitive performance in the standard intra-language scenario but also adapt and convert using speech utterances in an unseen language.

Index Terms— voice conversion, cross-lingual, speaker adaptation, transfer learning, text-to-speech

1. INTRODUCTION

The voice conversion (VC) system is used to convert speech of a certain speaker to speech of a desired target while retaining the linguistic content [1]. VC and text-to-speech (TTS) systems are two different systems functioning under different scenarios, but they share a common goal: generating speech with a target voice. If we define the TTS system as a synthesis system generating speech using linguistic instructions (in an abstract sense) obtained from written text [2], then the VC system can be defined as a synthesis system generating speech using linguistic instructions extracted from a reference utterance. This similarity in objective but difference in operation context makes VC and TTS complement each other and have their unique role in a spoken dialogue system. More specifically, as text input is easy to create and modify, TTS is capable of generating a large amount of speech automatically and cheaply. However it is difficult to generate speech when the desired linguistic instructions cannot be represented in the expected written form (e.g. when text is written in foreign languages). On the other hand, speech used as a reference input for VC is more time-consuming and expensive to produce, but the system can be straightforwardly extended to an unseen language.

A VC system is usually classified as parallel or non-parallel depending on the nature of the data available for development. The parallel VC system is developed using a parallel corpus containing pairs of utterance spoken by a deterministic source and target speakers. The parallel speech utterances of source and target are aligned using dynamic time warping (DTW) to create the training set. By using this training set, a mapping function is formulated to transform the acoustic features of one speaker to another [3]. Many methods have been proposed to model this transformation for parallel VC systems [1, 4, 5]. Parallel VC is developed with just data of source and target speakers which is one of its advantages. Acquisition of parallel speech data requires a lot more planning and preparation than acquisition of non-parallel speech, which in turn makes the former more expensive to develop, especially when we want a VC system with the voice of a particular speaker. Therefore, many approaches have been proposed to develop VC systems using a non-parallel corpus [6, 7]. For the conventional Gaussian mixture model (GMM) approach, non-parallel VC can be adapted from a pretrained parallel VC in the model space using the maximum a posterior (MAP) method [7, 8] or as interpolation between multiple parallel models [9]. For recent neural network approaches, a non-parallel VC can be trained by directly using an intermediate linguistic representation extracted from an automatic speech recognition (ASR) model [10, 11] or by indirectly encouraging the network to disentangle linguistic information from the speaker characteristics using methods like variational autoencoder (VAE) [12], generative adversarial networks (GAN) [13, 14] or some other techniques [15]. For both parallel and non-parallel VC, the systems usually change the voice but are unable to change the duration of the utterance. Many recent researches focused on converting speaking rate along with voices by using sequence-to-sequence models [11, 16, 17, 18], as speaking rate is also a speaker characteristic.

Recently Luong et al. proposed a speaker-adaptive TTS
model that could perform speaker adaptation with untranscribed speech using backpropagation algorithm [19]. We find that the proposed system can potentially be used for developing non-parallel VC as well. In this paper, we introduce a framework for creating a VC system using the untranscribed speech of a target speaker by bootstrapping from the speaker-adaptive TTS model proposed by Luong et al. [19]. Furthermore, we investigate the performance of our framework when it adapts and converts using speech utterances of an unseen language. The rest of paper is organized as follows. Section 2 describes our framework, Section 3 introduces different scenarios in which a VC can operate in an unseen language and their practical applications, Section 4 describes the experiment setup, Section 5 presents subjective results, and Section 6 concludes our findings.

2. BOOTSTRAPPING VOICE CONVERSION FROM SPEAKER-ADAPTIVE TTS

2.1. Multimodal architecture for speaker-adaptive TTS

We first summarize the speaker-adaptive TTS proposed in [19]. A conventional TTS model is essentially a function that transforms linguistic features \( x \), extracted from a given text, to the acoustic features \( y \), which could be used to synthesize speech waveform. A neural TTS model is a TTS model that uses neural networks as the approximation function for the \( x \) to \( y \) transformation. Neural networks are trained with backpropagation algorithm and can also be adapted to a new unseen domain with backpropagation when the adaptation data is labeled. In the context of TTS, an acoustic model can easily be adapted to voices of unseen speakers if the adaptation data is transcribed speech but it is not as straightforward when the adaptation is untranscribed. Many techniques have been proposed to perform adaptation with untranscribed speech by avoiding backpropagation and extracting a certain speaker representation from an external system instead [20, 21]. However, this forward-based approach limited the performance of the adaptation process [19].

Luong et al. [19] proposed backpropagation-based adaptation method using untranscribed speech by introducing a latent variable \( z \), namely latent linguistic embedding (LLE), which presumably contains no information about the speakers. The conventional acoustic model then splits into a speaker-dependent (SD) acoustic decoder \( \text{Dec} \) and a speaker-independent (SI) linguistic encoder \( \text{LEnc} \). The acoustic decoder is defined by its SI parameter \( \theta^{\text{core}} \) and SD parameter \( \theta^{\text{spk}(k)} \), one for each \( k \)-th speaker in the training set while the linguistic encoder is defined by its SI parameter \( \phi^{L} \). The TTS model can be described as follows:

\[
\begin{align*}
  z^{L} &\sim \text{LEnc}(x; \phi^{L}) = p(z|x) \tag{1} \\
  \hat{y}^{L} & = \text{Dec}(z^{L}; \theta^{\text{core}}, \theta^{\text{spk}(k)}) \tag{2}
\end{align*}
\]

With this new factorized model, if we want to adapt the acoustic decoder to unseen speakers using speech, we train an auxiliary acoustic encoder \( \text{AEnc} \) (defined by its SI parameter \( \phi^{A} \)) to use as a substitute for the linguistic encoder:

\[
\begin{align*}
  z^{A} &\sim \text{AEnc}(y; \phi^{A}) = q(z|y) \tag{3} \\
  \hat{y}^{A} & = \text{Dec}(z^{A}; \theta^{\text{core}}, \theta^{\text{spk}(k)}) \tag{4}
\end{align*}
\]

The acoustic encoder is only used to perform the unsupervised speaker adaptation in [19]; however the network created by stacking the acoustic encoder and acoustic decoder is essentially a speech-to-speech stack that potentially could be used as a VC system. This is the motivation for us to develop a VC framework based on this model.

**Fig. 1.** Training the initial speaker-adaptive TTS model.

**Fig. 2.** Transferring to VC and adapting to the target speaker.
2.2. Latent Linguistic Embedding for many-to-one voice conversion

By using the speaker-adaptive TTS model described above, we propose a three-step framework to develop a VC system for a target speaker:

1) train the initial TTS model: The first step is to train the TTS model, it is identical to the training mode defined in Section IV-B in [19]. The data required for this step is a multi-speaker transcribed speech corpus. In this step, we need to jointly train all modules using a deterministic loss function:

$$\text{loss}_{\text{train}} = \text{loss}_{\text{main}} + \beta \text{loss}_{\text{tie}}$$  \hspace{1cm} (5)

loss$_{\text{main}}$ is the distortion between output of the TTS stack and the natural acoustic features, and loss$_{\text{tie}}$ is the KL divergence (KLD) between the output of the linguistic encoder and output of the acoustic encoder. The setup is almost the same as described in [19]. One small difference is that we use the mean absolute error (MAE) as the distortion function instead of the mean square error as shown in Fig. 1. In this step, the acoustic encoder is trained to transform acoustic features to the linguistic representation LLE while the acoustic decoder is trained to become a good initial model for speaker adaptation.

2) transfer to VC and adapt to the target: For building a VC system we only need the acoustic encoder and the acoustic decoder trained previously; the linguistic encoder is discarded as it is no longer needed. The acoustic encoder and decoder make a complete speech-to-speech network. To have a VC system with a particular voice we adapt the acoustic decoder using the available speech data of the target. This is identical to the unsupervised adaptation mode described in Section IV-B in [19]. More specifically, we removed all SD parameters $\theta_{\text{spk}}(k)$ and then fine-tuned remaining parameter $\theta_{\text{core},(r)}$ to the $r$-th target speaker by minimizing the loss function:

$$\text{loss}_{\text{adapt}} = L_{\text{MAE}}(\hat{y}^A, y)$$  \hspace{1cm} (6)

Slightly different to [12], we removed the standard deviation $\sigma$ from the acoustic encoder as illustrated in Fig. 2. Simply speaking instead of sampling $z^A$ from a distribution (Equation 3), $z^A$ will always take the mean value instead. We found this change slightly improves the performance of the adapted model in the converting step.

3) converting speech of arbitrary source speakers: The adapted VC system can then be used to convert speech of arbitrary source speakers to the target. Our system does not need to train on or adapt to the speech data of source speakers, although doing so might help the performance.

2.3. Related works

While the original VC is narrowly defined as a function that directly maps acoustic units of a source to a target speaker [3, 22], recent VC systems have usually been modeled with an intermediate linguistic representation, either explicitly [10] or implicitly [24] due to the rapid development of representation learning and disentanglement ability of neural networks. This approach has further reduced the theoretical difference between VC and TTS. Motivated by the same observation, Mingyang Zhang et al. [24] proposed a method to jointly train a TTS and a parallel VC system using a shared sequence-to-sequence model. The proposed framework improves the stability of the generated speech but does not add any benefits in term of data efficiency, as parallel speech data of the source and target speakers is still a requirement. Recently, Jing-Xuan Zhang et al. [18] have introduced a non-parallel sequence-to-sequence voice conversion system that has a procedure that is similar to our: the initial model is trained with a TTS-like network to help disentangle linguistic representation and the model then adapted to the source and target speakers. However, the adaptation step of the framework proposed in [18] requires both source and target speaker speech as well as their transcripts, which increases the data demand for building a VC system with a particular voice.

Despite being different in motivation and procedure, our framework has the same data efficiency as the models using phonetic posteriorgrams (PPG) proposed by Sun et al. [10]. The LLEs in our setup play a similar role to the PPGs, but are very different in terms of characteristics. While PPGs are the direct output of an ASR network with an explicit interpretation, LLEs are latent variables and guided by a TTS network. The efficiency in terms of data usage is not only lowering the requirement for building a VC system but also opening up the possibility of solving other interesting tasks. Creating and using the VC system with a low-resource unseen language is one example [25].

3. USING VOICE CONVERSION SYSTEM IN AN UNSEEN LANGUAGE

The proposed three-step framework have an interesting characteristic which is the asymmetric nature in terms of data requirements for each step. While the TTS training step requires a transcribed multi-speaker corpus, the adaptation step only requires a small amount of untranscribed speech from the target speaker. Moreover the model does not need to train on data of source speakers and hypothetically can convert utterances of arbitrary speakers out of the box. We could take advantage of these characteristics to build VC systems for low-resource languages. For our experiments, we define the abundant language as language with transcribed speech corpus and used to train the initial TTS model (seen); in our case it is English (E). The low-resource language is a language whose data does not include transcripts and is not used in the TTS step (unseen); in our case it is Japanese (J). We first define multiple scenarios in which our framework can interact with an unseen language. These scenarios are dictated by the language used in the adaptation and conversion steps:
Fig. 3. The asymmetric nature of data requirements for each step of the proposed framework.

(a) EE-E: We adapt the pretrained English model to an English speaker and using it to convert English utterances. Generally speaking the entire framework from start to finish operates within a single language. This is the typical intra-language scenario of a voice conversion system.

(b) EE-J: We adapt the pretrained English model to an English speaker but use it to convert Japanese utterances. Generally speaking, the VC system is used to generate speech of linguistic instructions that have not been seen. The ability to generate speech for content that is difficult to represent in the expected written form (a foreign language in this case) is one advantage of VC over TTS. This scenario is referred as cross-language voice conversion [26, 27].

(c) EJ-E: We adapt the pretrained English model to a Japanese speaker and use it to convert English utterances. In this scenario we want to build a VC system in the abundant language; however, the speech data of the target speaker is only available in a low-resource unseen language. This is sometime referred to as cross-lingual voice conversion [28, 29], however we call it cross-language speaker adaptation to distinguish it from EE-J. Unlike other scenarios involving the unseen language, cross-language speaker adaptation is relevant for both VC [28] and TTS [30, 25]. Even though it is not evaluated in this paper, this scenario is also the unsupervised cross-language speaker adaptation scenario of the TTS system proposed in [19].

(d) EJ-J: We adapt the pretrained English model to a Japanese speaker and use it to convert Japanese utterances. In this scenario we essentially bootstrap a VC system for a low-resource language from a pretrained model of an abundant one. The written form of the target language is not used in the training, the adaptation or the conversion stages. This scenario is sometime referred as text-to-speech without text, which is the main topic of the Zero Resource Speech Challenge 2019 [31]. Even though our scenario has the same objective as the challenge, the approach is a little different. The participant of the challenge are encouraged to develop intra-language unsupervised unit discovery methods, which are more difficult [32, 33, 34]. Our framework is bootstrapped from an abundant language, which is a more practical approach [35, 36].

Fig. 4. Different scenarios of using a non-parallel voice conversion system with unseen language. In this paper, English plays the role of the abundant language (seen) while Japanese plays the role of the low-resource language (unseen).

4. EXPERIMENT

4.1. Dataset

We used 25.6 hours of transcribed speech from 72 English speakers to train the initial speaker-adaptive TTS model and the initial SI WaveNet vocoder. The data is a subset of the VCTK speech corpus [37]. To validate the proposed method, we reenact the SPOKE task of the Voice Conversion Challenge 2018 (VCC2018) [38]. We build the VC system for 4 target speakers (2 males and 2 females) using 81 utterances per person. We then evaluate the SPOKE task using the speech of 4 source speakers (2 males and 2 females); each speaker contribute 35 utterances. Even though the task provided training data for the source speakers, we did not use the training data in our experiments as our model can convert speech of an arbitrary source speaker.

To test the performance of the proposed system in the context of unseen language as described in Section 3 we use 2 in-house bilingual speakers (1 male and 1 female) from VCC2018 as the English source speakers and add 2 native Japanese speakers (1 male and 1 female) as the Japanese source speakers.
Our configuration imitates the setup described in [19] as closely as possible. The linguistic feature is a 367-dimensional vector containing various information that is deemed to be useful for English speech synthesis. The acoustic feature is the 80-dimensional mel-spectrogram. One difference compared with [19] is that all speech data is resampled to 22050 Hz (which is the sampling rate of the VCC2018 dataset) before it is used to extract acoustic features. WaveNet vocoder is also trained on the 22050 Hz waveform that has been quantized using 10-bit u-law.

The neural network used for the speaker-adaptive acoustic model has the same structure as in [19] with the size of 40 dilated causal layers and it is conditioned on a natural mel-spectrogram. The WaveNet vocoder is fine-tuned for each target speaker using their respective available training speech data.

4.2. Model configuration

Our configuration imitates the setup described in [19] as closely as possible. The linguistic feature is a 367-dimensional vector containing various information that is deemed to be useful for English speech synthesis. The acoustic feature is the 80-dimensional mel-spectrogram. One difference compared with [19] is that all speech data is resampled to 22050 Hz (which is the sampling rate of the VCC2018 dataset) before it is used to extract acoustic features. WaveNet vocoder is also trained on the 22050 Hz waveform that has been quantized using 10-bit u-law.

The neural network used for the speaker-adaptive acoustic model has the same structure as in [19] with the size of 40 dilated causal layers and it is conditioned on a natural mel-spectrogram. The WaveNet vocoder is fine-tuned for each target speaker using their respective available training speech data.
their similarity result trend is the same as their quality result in Fig. 8. For the four systems that convert Japanese speech, score than EE-J. The result for the similarity test is illustrated standard scenario JJ-J; as expected EJ-J has slightly better converting scenarios, EJ-J and EE-J, are a lot worse than the they should not be compared directly. The cross-language score than the English counterpart EE-E, although technically standard Japanese system JJ-J achieved a relatively better which is only judged 1400 times for each measurement.

Each scenario is judged 2800 times for quality and another 2800 times for similarity except the reference natural speech, which is prepared to contain all source and target pairs. In summary, each scenario is judged 544 times for quality and another 544 times for similarity. The quality and similarity results are illustrated in Fig. 6 and are statistically significant between all scenarios. The cross-language speaker adaptation EJ-E is generally worse than EE-E as one would expect.

5.2. Cross-language speaker adaptation

Next, we evaluate our system in the scenario of cross-language speaker adaptation (EJ-E). For this task we develop a system using the speech data of the bilingual target speakers. We compare EJ-E with EE-E, which is the standard intra-language of English, and a reference system NA-E, which is of hold-out natural English utterances.

The native English speakers that participated in the survey for previous task are asked to evaluate the cross-language speaker adaptation task as well. In summary, each scenario is judged 544 times for quality and another 544 times for similarity. The quality and similarity results are illustrated in Fig. 6 and are statistically significant between all scenarios. The cross-language speaker adaptation EJ-E is generally worse than EE-E as one would expect.

5.3. Voice conversion for low-resource language

Finally, we test the performance of our system when using it to convert speech of low-resource language. This consists of the EE-J and EJ-J scenarios. We use JJ-J, the abundant language scenario of Japanese, as the upper-bound and NA-J, the hold-out natural Japanese utterances, as the reference. We prepare a similar listening test to that of previous tasks. A total 148 native Japanese participated in our test; no participant is allowed to do more than 10 sessions. Each session is prepared to contain all source and target pairs. In summary, each scenario is judged 2800 times for quality and another 2800 times for similarity except the reference natural speech, which is only judged 1400 times for each measurement.

The result of the quality test is illustrated in Fig. 7. Our standard Japanese system JJ-J achieved a relatively better score than the English counterpart EE-E, although technically they should not be compared directly. The cross-language converting scenarios, EJ-J and EE-J, are a lot worse than the standard scenario JJ-J; as expected EJ-J has slightly better score than EE-J. The result for the similarity test is illustrated in Fig. 8. For the four systems that convert Japanese speech, their similarity result trend is the same as their quality result trend. For the similarity test, we also included two extra systems that produce English speech, EE-E and NA-E. In these two cases the listener would listen to an English utterance contributed by EE-E or NA-E and a natural Japanese utterance of the same speaker (as the target speakers are bilingual) and judge their similarity. Interestingly the similarity result of NA-E is a lot worse than that of NA-J even though they both contain natural speech spoken by the same speaker in real life. This suggests that utterances spoken by the same speaker in different languages might not have consistent characteristic. Or problem might be that is it difficult for listeners to evaluate speaker similarity in cross-language scenario, especially when they are not fluent in both.

Even though the performance in the unseen language scenarios is not as good as the standard scenario, the subjective results have shown the ability of using our system for inter-language tasks and establishing a preliminary result as well as a solid baseline for future improvements.

6. CONCLUSION

In this paper, we have presented a methodology to bootstrap a VC system from a pretrained speaker-adaptive TTS model. By transferring knowledge learned previously by the TTS model, we were able to significantly lower the data requirements for building the VC system. This in turn allows the system to operate in a low-resource unseen language. The subjective results show that our VC system achieves competitive performance compared with existing methods. Moreover, it can also be used for cross-language voice conversion and cross-language speaker adaptation. While the performance in these unseen language scenarios are not as good, all experiments in this work are conducted under the assumption of minimal available resources. Our future work includes taking advantage of the available additional resources such as a multi-lingual corpus to further improve the robustness of the VC system.

1Speech samples are available at https://nii-yamagishilab.github.io/sample-vc-bootstrapping-tts/
7. REFERENCES


