11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Preliminary study on using vector quantization latent spaces for TTS/VC systems with consistent performance

Hieu-Thi Luong[1], *Junichi Yamagishi*[1]

[1]National Institute of Informatics, Tokyo, Japan

{luonghieuthi,jyamagis}@nii.ac.jp

Abstract

Generally speaking, the main objective when training a neural speech synthesis system is to synthesize natural and expressive speech from the output layer of the neural network without much attention given to the hidden layers. However, by learning useful latent representation, the system can be used for many more practical scenarios. In this paper, we investigate the use of quantized vectors to model the latent linguistic embedding and compare it with the continuous counterpart. By enforcing different policies over the latent spaces in the training, we are able to obtain a latent linguistic embedding that takes on different properties while having a similar performance in terms of quality and speaker similarity. Our experiments show that the voice cloning system built with vector quantization has only a small degradation in terms of perceptive evaluations, but has a discrete latent space that is useful for reducing the representation bit-rate, which is desirable for data transferring, or limiting the information leaking, which is important for speaker anonymization and other tasks of that nature.

Index Terms: voice cloning, text-to-speech, voice conversion, vector quantization, variational autoencoder

1. Introduction

When it comes to text-to-speech (TTS) tasks, the deep learning approach has several advantages over the conventional approaches, such as its simple structure and the ability to scale with large data [1, 2]. These characteristics are key for pushing the performance of speech synthesis systems and machine learning systems in general. Recent works have shown that a sequence-to-sequence TTS system [1, 3] trained with a large transcribed speech corpus can synthesize speech with high naturalness directly from text input instead of going through several sub-systems. While such systems provide a high performance and a straightforward solution for TTS, many researchers have shifted their focus to more elaborate systems that aim to give some of the control back to human users [4, 5]. For example, Shen et al. [6] replaced the attention module with a duration prediction to create a more resilient sequence-to-sequence TTS model and enable the ability to control the spacing of generated speech, while Liu et al. [7] explicitly integrated emphatic code into the model so that users can control the emphasis by changing duration, intonation, and energy.

For voice conversion (VC), recent deep learning based systems are formulated around the ability to disentangle speaker and linguistic information of a neural network by using an information bottleneck structure to force the model to learn useful representations [8, 9, 3]. This information bottleneck structure can simply be a layer with a few units [10], a variational autoencoder (VAE) model with its encoder's output regularized to approximate a normal distribution [11, 12], or a jointly trained discrete latent space through vector quantization [13, 14, 9]. In these works, the common hypothesis is that using a certain network structure can help train a representation that takes on information and/or properties that are useful for the task at hand.

Previously, we proposed NAUTILUS [15], a versatile voice cloning system, that is a fusion of TTS and VC. By carefully designing the shared and the exclusive components, NAUTILUS can utilize them to perform elaborate tasks such as cloning unseen voices using untranscribed speech. It also has a consistent performance when switching between TTS and VC. These properties are the result of a unified and robust linguistic latent space achieved by the joint training and the VAE-like structure. However, one may want to use other methods to shape the latent space for many different purposes. Specifically, if we can train a discrete latent space [16, 17] instead of a continuous one, it will be useful for many applications. For example, a low bit-rate representation [13, 18] is ideal for a client-server VC system in which the speech encoder is stored in the client device while the speech decoder is not. Alternatively, by using the vector quantization bottleneck, we can limit the information getting through the speech encoder, which is important for tasks such as speaker anonymization [19, 20] as it helps reduce the leaking of speaker identity through temporal patterns. In this work, we investigate the use of vector quantization variational autoencoder (VQVAE) [17] components to model the linguistic latent space of the NAUTILUS system. In addition to conducting experiments to clarify its effect on subjective evaluations, we discuss how different types of assumption about the latent spaces are useful for different scenarios. We describe the basics of the voice cloning framework with the vector quantization components in Section 2 and discuss our motivation in Section 3. Section 4 lays out the experiment conditions, and Section 5 presents the subjective evaluation results. We conclude in Section 6 with a brief summary of our findings and mention of future works.

2. Vector Quantization Latent Space for Voice Cloning

We adopt the basic concepts of the voice cloning framework proposed in our previous publications [15] and replace the VAE-based encoders with a VQVAE-based counterpart for this work. Readers may want to refer to the original study [15] for more context. Briefly, our voice cloning system is a unified system of TTS and VC, and thanks to this fusion it has the capacity to clone new voices using untranscribed speech. The proposed VQVAE-based system, called NAUTILUS-VQ, is illustrated in Fig. 1. The only difference from the original [15] is the way the text and speech encoders are set up as shown in Fig. 2. More specifically, the vector quantization bottleneck transforms the continuous latent feature z, emitted by the text or speech encoder, into a discrete latent feature q using the jointly trained codebook $\boldsymbol{e}_k, k \in 1...K$, with $q = e_k$ where $k = argmin_j ||z - e_j||$. The speech decoder then consumes q,

10.21437/SSW.2021-24

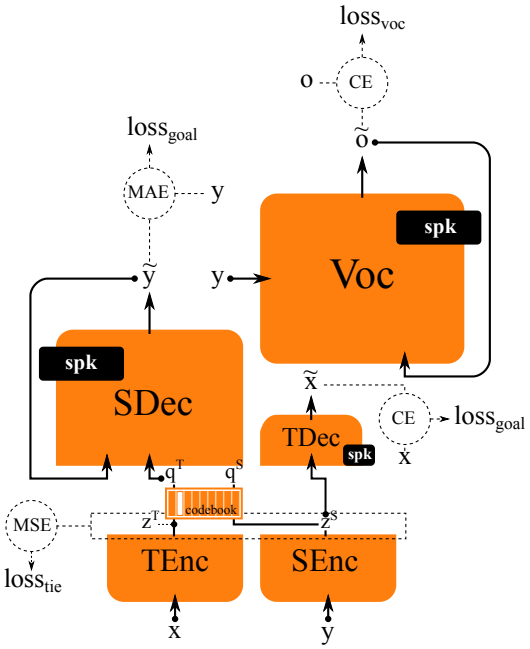
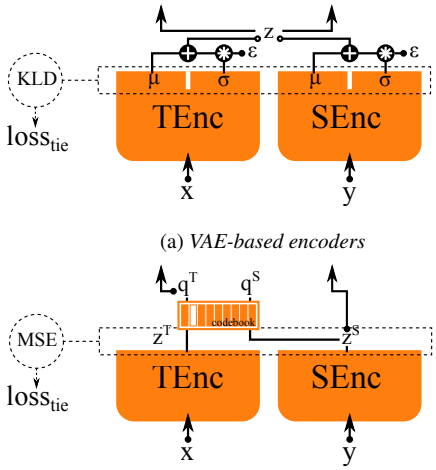


Figure 1: *The modified NAUTILUS-VQ system has a similar structure to the original system, which includes a text encoder ($TEnc$), a speech encoder ($SEnc$), a text decoder ($TDec$), a speech decoder ($SDec$), and a neural vocoder (Voc). The jointly trained codebook is the new addition. $\boldsymbol{x}$ is phoneme, $\boldsymbol{y}$ is acoustic, $\boldsymbol{o}$ is waveform, $\boldsymbol{z}$ is continuous latent feature and $\boldsymbol{q}$ is the quantized latent feature. The term loss_{goal} is a placeholder, depending on the encoder/decoder combination — it can be loss_{tts} (Text-to-speech: $TEnc$→codebook→$SDec$), loss_{sts} (Speech-to-speech, STS: $SEnc$→codebook→$SDec$), loss_{stt} (Speech-to-text, STT: $SEnc$→$TDec$), or loss_{ttt} (Text-to-text: $TEnc$→$TDec$) . The loss functions used in training and adaptation are mean absolute error (MAE), mean squared error (MSE) and cross entropy (CE).*



(a) *VAE-based encoders*

(b) *VQVAE-based encoders*

Figure 2: *The VAE (original) and VQVAE (proposed) setups for the encoders. The VAE-based encoders output mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ and then sample the latent feature $\boldsymbol{z}$ using an $\boldsymbol{\epsilon}$ value drawn from a normal distribution. Kullback-Leibler divergence (KLD) is used as loss_{tie} in this setup. The VQVAE-based encoders output continuous latent feature $\boldsymbol{z}$ that is then quantized into the discrete feature q by using the jointly trained codebook. MSE is used for loss_{tie} in this case.*

instead of z, to reconstruct the acoustic feature y that is used to synthesize speech waveform o.

2.1. Train the initial model

First, we need to jointly train the text/speech encoders/decoders and the codebook in a supervised fashion by using a large-scale transcribed multi-speaker speech corpus and optimizing a designated loss:

$$\text{loss}_{train} = \text{loss}^{tts}_{train} + \alpha_{sts}\ \text{loss}^{sts}_{train} + \alpha_{stt}\ \text{loss}_{stt} + \beta\ \text{loss}_{tie}\ . \quad (1)$$

Please see the caption of Fig. 1 for subscripts of each term. The basic structure of the training loss is not much different from the original [15], but, due to the vector quantization components, the details are a little more complex. Specifically $\text{loss}^{tts}_{train}$ is similar to a typical VQVAE setup [17]:

$$\text{loss}^{tts}_{train} = \text{loss}_{tts} + \delta_{VQ}\ \text{loss}^{T}_{VQ} + \delta_{C}\ \text{loss}^{T}_{C}\ . \quad (2)$$

where $\text{loss}_{tts} = ||\tilde{y}^T - y||$ is the reconstruction loss of the acoustic feature, $\text{loss}^{T}_{VQ} = ||sg(z^T) - q^T||^2_2$ is the codebook training loss in response to the text input, x, from the text encoder, and $\text{loss}^{T}_{C} = ||z^T - sg(q^T)||^2_2$ is the text encoder commitment. The operator $sg()$ indicates the stop gradient operation.

Similarly, we have the optimization loss for the STS stack that handles the speech input:

$$\text{loss}^{sts}_{train} = \text{loss}_{sts} + \delta_{VQ}\ \text{loss}^{S}_{VQ} + \delta_{C}\ \text{loss}^{S}_{C}\ . \quad (3)$$

Unlike the VAE-based setup [15] which used Kullback-Leibler (KL) divergence to "tie" the encoders' outputs, the VQVAE-based system uses MSE as the latent tying loss:

$$\text{loss}_{tie} = ||sg(z^T) - z^S||^2_2\ . \quad (4)$$

Note that we stop the gradient on the text-encoded latent feature, which basically creates an asymmetric tied-layer loss instead of the symmetric KL divergence function as in the original [15]. A multi-speaker WaveNet vocoder, unchanged from the original, is separately initialized using the same corpus:

$$\text{loss}'_{train} = \text{loss}_{voc}\ , \quad (5)$$

The speech decoder, text decoder, and neural vocoder contain speaker dependent (SD) components, which are just one-hot vectors representing speakers in the training set. These SD components will be removed in the voice cloning steps along with the text decoder, which is only included as an auxiliary phone classification regularizer [15].

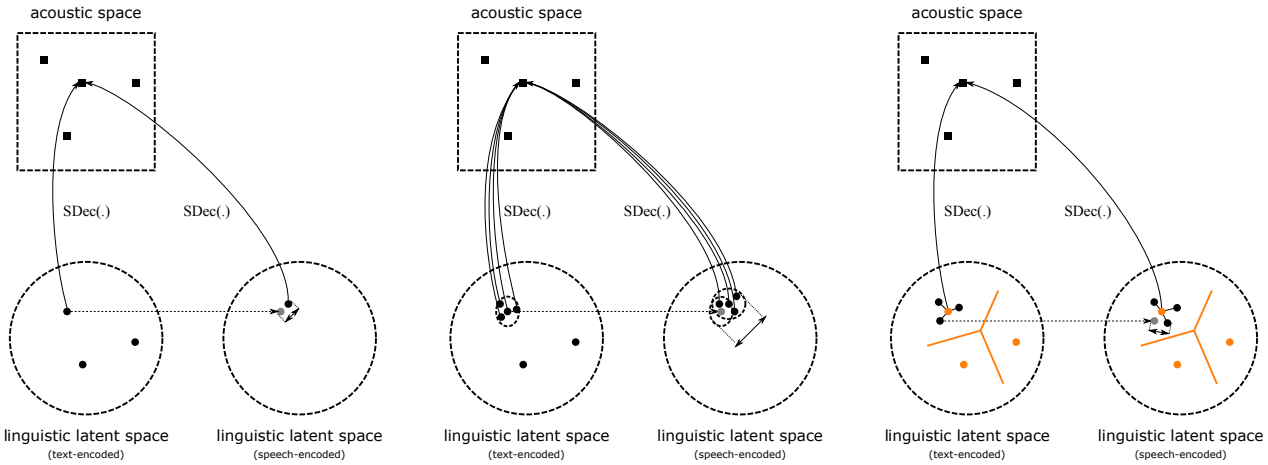
2.2. Clone the target voice

Given the untranscribed speech of an unseen speaker, we adapt the initial model to generate speech with the voice of the new target, using either the TTS or VC interface.

2.2.1. Step 1 - Adaptation

After removing all SD components, we use the STS stack ($SEnc$→codebook→$SDec$) to fine-tune the speech decoder while keep other modules immutable:

$$\text{loss}_{adapt} = \text{loss}_{sts}\ , \quad (6)$$



(a) *Standard* (b) *VAE-based* (c) *VQVAE-based*

Figure 3: *Different ways the linguistic latent spaces are setup and the methods used to enforce the consistency between them.*

The neural vocoder is also adapted using the same strategy (removing SD components, fine-tuning the rest):

$$\text{loss}'_{adapt} = \text{loss}_{voc} \ . \tag{7}$$

As this step is similar to the original framework, reader can refer to Fig. 2a in [15] for details.

2.2.2. *Step 2 - Welding*

The adapted model obtained in step one is already capable of generating speech with the target voice. However, we want to increase the speech decoder and the neural vocoder compatibility by jointly tuning them using the speech to waveform stack ($SEnc \rightarrow$codebook$\rightarrow SDec \rightarrow Voc$):

$$\text{loss}_{weld} = \text{loss}_{sts} + \gamma \ \text{loss}_{voc} \ . \tag{8}$$

The loss_{sts} is included to maintain the acoustic space for the autoregressive speech decoder (see Fig. 2b in [15]).

2.2.3. *Step 3 - Inference*

After the previous steps, the adapted model can generate speech with the voice of the target. The TTS stack and the neural vocoder form a TTS interface that transforms phoneme sequences into a speech waveform, while the STS stack and the neural vocoder form a VC interface that converts utterances spoken by arbitrary source speakers into speech of the same content but with the voice of the target (Fig. 2c in [15]).

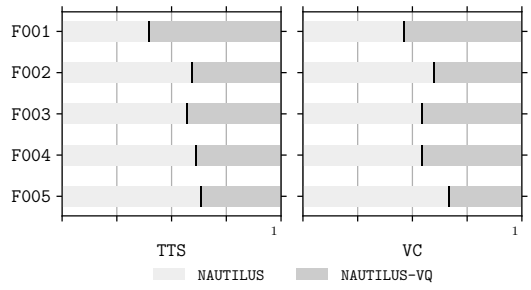
3. Shaping the linguistic latent spaces

Our voice cloning system functions on the assumption of a robust and consistent linguistic latent space. Briefly, we want the latent linguistic embedding (LLE) to contain linguistic information, which is useful for speech reconstruction, but none of the speaker information. Moreover, the consistency between the TTS and VC interfaces is dictated by the consistency between the text-encoded and speech-encoded latent spaces. In other words, the proposed system achieves a perfect consistency if its text and speech encoders produce an identical LLE sequence given a sentence input and a spoken utterance of the same content. Note that, for many practical applications [21], it may not even be desirable to achieve such consistency as it eliminates the ability to synthesize speech content that cannot be represented in written form. However, establishing a straightforward goal about consistency helps to simplify the analysis.

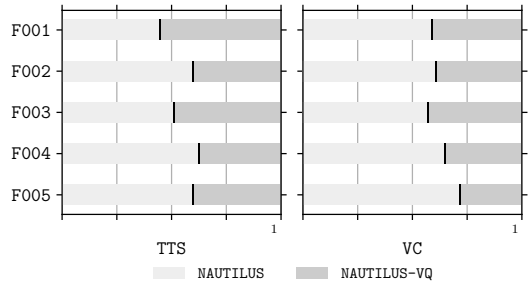
Table 1: *Japanese target speakers for voice cloning task*

Speaker	Gender	Quantity	Duration
F001	female	483 utt.	45.0 min
F002	female	481 utt.	44.4 min
F003	female	484 utt.	47.4 min
F004	female	468 utt.	40.8 min
F005	female	485 utt.	47.6 min
XL10	female	10 utt.	55 s
		125 utt.	10.9 min
		500 utt.	44.5 min
		2000 utt.	2.9 h
		8750 utt.	12.9 h

We can simply use standard latent features [22] as LLEs and assume the text and speech encoders produce identical features when consuming different modalities of the same content, and so we optimize the consistency between them by minimizing the distance between the two latent points [23] as shown in Fig. 3a. The choice of distance function is another decision that could affect the performance [24], but in practice most use Euclidean distance for its simplicity [25]. The flaws of this assumption are the one-to-many relation of text and speech and the scarcity nature of data, which make it difficult to train a robust and consistent latent space. To address this problem, we utilized the VAE-based encoders (Fig. 3b) in our previous works [15]. This modification provides two key benefits: 1) the speech decoder is trained in a denoising fashion due to the sampling process, which can be interpreted as an artificial data argumentation, and 2) we can use a density-wise instead of a point-wise function to connect the text and speech encoders which helps with the consistency. However, it has the common drawback of the VAE model [26] which is the average-ness of the generated features [3]. Therefore, in this paper, we investigate the VQVAE-based modification that has discrete latent spaces, as shown in Fig. 3c. The hypothesis is that the discrete features will allow the speech decoder to learn fine-grain details. Moreover, it has several useful traits as mentioned in previous sections.

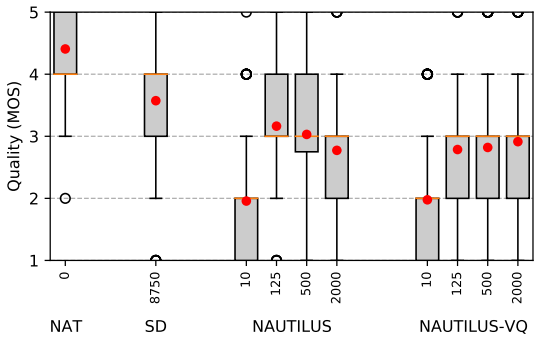


(a) *Quality*

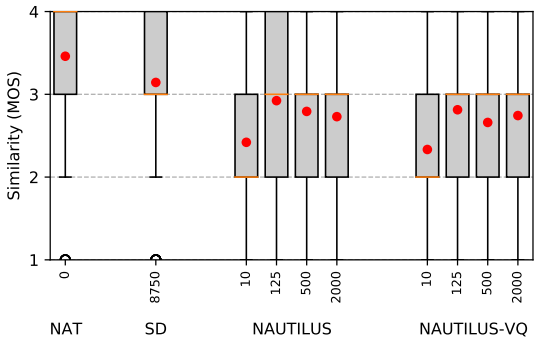


(b) *Similarity*

Figure 4: *Subjective evaluations on quality and speaker similarity between NAUTILUS and NAUTILUS-VQ on voice cloning task for TTS and VC.*



(a) *Quality*



(b) *Similarity*

Figure 5: *Subjective evaluations on quality and speaker similarity of the TTS unsupervised speaker adaption using varying amount of untranscribed speech data of speaker XL10.*

4. Experiments

4.1. Model and training configurations

For the experiments, we compared the performances of the original NAUTILUS system [15] and the new system with vector quantization modification. The network architecture of the original was unchanged from the previous publication, and consists of many layers of causal and non-causal dilated convolution layers. Readers can refer to Fig. 4 and Sec. IV of [15] for details. The NAUTILUS-VQ system also adopts this architecture but with several changes to reflect the vector quantization components. Specifically, the encoders directly output 64-dimensional latent vectors instead of means and standard variances. The 160-code jointly trained codebook was used to transform these continuous features into discrete ones. We chose this size for the codebook because it produces a relatively reliable performance based on several test-runs and relevant publications pertaining to VQVAE [14, 27]. For the hyperparameters, we set $\alpha = 0.1$, $\beta = 0.25$, $\gamma = 0.01$, the same as in [15], and $\delta_C = 1.0$, $\delta_{VQ} = 0.25$, as based on relevant works [17, 14].

4.2. Speech data

Previously, we conducted experiments with English as the target language [15]. In this work, we used Japanese to test our methodology under a new condition. Specifically, several native female Japanese speakers, as listed in Table 1, were selected as the target speakers. The Japanese model was first initialized on a large-scale low-quality transcribed speech corpus with a diverse linguistic content. We used ∼236 hours of speech (978 speakers) from the 16 khz Corpus of Spontaneous Japanese (CSJ) [28] for this purpose. Then, we fine-tuned it on a quality-controlled transcribed speech corpus for the desired sampling rate. We used ∼134 hours of speech (235 speakers) from an in-house 24 khz Japanese Voice Bank Corpus for this step. The same policy was applied for the training of both NAUTILUS and NAUTILUS-VQ, and it is similar to the one we used when we trained the English models [15]. The well-trained modules were used to adapt to the target speakers using their untranscribed speech data. For the first scenario, five speakers with about 45 minutes of speech were used to compare the performances of NAUTILUS and NAUTILUS-VQ. For the second scenario, a different speaker, XL10, was used to investigate the performances when adapting with different amounts of data. This speaker was chosen because she was included in our previous experiments using the conventional TTS system [29].

5. Evaluations

5.1. Vector quantization latent spaces for voice cloning

We first compare the original and the VQ-based systems on the voice cloning task: specifically their perceptive evaluations for TTS and VC[1]. We used about 45 minutes of untranscribed speech of the first five target speakers in Table 1 for this scenario. This amount of data was more than our previous English experiments [15], which were conducted with just five or ten minutes of speech. We used a crowdsourcing service to conduct the survey. A total of 241 native speakers, each of whom did one to five sessions, participated. Listeners were asked to pick the preferred sample between two presented in terms of either quality or speaker similarity (which includes a reference sample). The results are shown in Fig. 4. In total, each speaker/system/task was judged 300 times. Interestingly, the NAUTILUS-VQ system had worse results for most speakers except F001. As most of the differences between the two

[1]Samples are available at `https://nii-yamagishilab.github.io/sample-preliminary-nautilus-vq/`

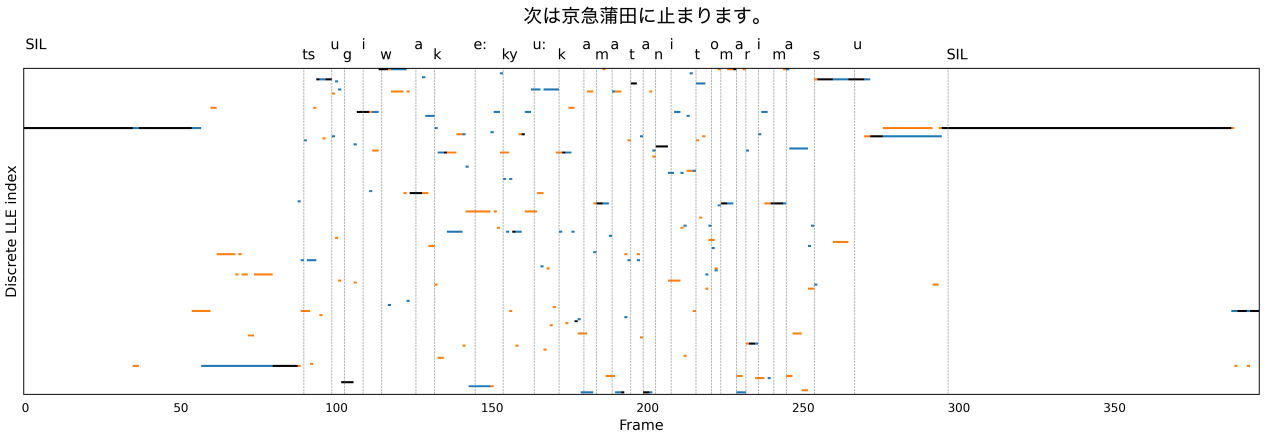


Figure 6: *Examples of the 160-code discrete LLE sequences. One utterance of a source speaker was used to generate the speech-encoded LLE (orange), while text (phoneme) and alignment information extracted from the same utterance was used to generate the text-encoded LLE (blue). The color black indicates the overlap of the speech-encoded and text-encoded LLE sequences, which covers 54.41% of this particular example utterance.*

systems were marginal, we conclude that the proposed vector quantization latent space can be used for voice cloning but has a small degradation in quality.

5.2. TTS speaker adaptation with different amount of untranscribed speech data

Next, we investigate the performances of both NAUTILUS and NAUTILUS-VQ for TTS unsupervised speaker adaptation with varying amount of adaptation data. For this scenario, we use speaker XL10, with whom we have more than 13 hours of speech data. Previously, we found that with this amount of data, the SD system of this speaker does not seem to benefit from the joint training with augmented data from other speakers [29]. In addition to the natural utterances (NAT) and generated utterances from our voice cloning systems, we included generated utterance from the conventional TTS system trained on 12.9 hours of transcribed speech of XL10. Basically, we reused the SD system in [29] as the upper bound for this experiment. The same listeners who participated in the first scenario survey were asked to do this one as well. Specifically, they were asked to judge the naturalness of a speech sample in a typical 5-point scale mean opinion score (MOS) question, and the likelihood that two presented samples were spoken by the same person on a 4-point scale. The results are shown in Fig. 5. As expected, none of the TTS systems were as good as natural speech, but the speaker similarity was not too far behind. Second, none of our voice cloning systems were as good as the SD baseline, which is not surprising as SD was trained on 12.9 hours of transcribed speech [29] while our systems cloned voices with only a small amount of untranscribed utterances. Third, between our systems, NAUTILUS-VQ has worse results than NAUTILUS in most data points, but not significantly so. Finally, the most interesting results were the performances when adapting to varying amounts of data, where we found that the performance of the NAUTILUS system seemed to peak at 125 utterances. These findings demonstrate the potential of our voice cloning system but at the same time reveal its remaining limitations.

5.3. Visualization of the discrete LLE sequences

One advantage of our speech synthesis is its ability to use as TTS and VC while maintaining a relatively consistent performance when switching between the two. Figure 6 shows the LLE sequences generated from text input and an utterance spoken by a source speaker not included in the training and adaptation stages. As the discrete LLE is forced to be one of 160 jointly trained vectors, it is easier and more intuitive to evaluate the consistency between the text-encoded and speech-encoded LLE sequences compared with the continuous representation (Fig. 8 in [15]). Figure 6 shows the discrete LLE sequences generated by the text and speech encoders using text and speech of the same content. For a perfectly consistent TTS/VC system, we expect the two sequences to be perfectly matched but it was not the case as seen in Fig. 6. Most of the overlap occurred at the start and the end of the utterance, which is the silence phoneme, but not much elsewhere. Overall, the LLE sequences were sparse and fragmented. For further improvement, focusing on condensing the latent space and stabilizing the text-encoded and speech-encoded LLE sequences would be a good direction.

6. Conclusion

In this paper, we investigated the feasibility of using VQVAE-based components to train a discrete latent linguistic embedding for a consistent performance TTS/VC system. While the perceptive evaluations showed that the proposed NAUTILUS-VQ system is not as good as the original system, having these different approaches to model the linguistic latent spaces is handy for many practical reasons. Understanding the dynamics of different methods is also important for the development of a more sophisticated speech synthesis system that can solve more complex and elaborate tasks, such as controlling speaking style [5], denoising TTS [30], or generating audio other than speech [31]. As VQVAE is just one way to model a jointly trained discrete latent space, other methods [16, 32] or assumptions [14, 33] about the nature of the latent space may lead to different results and have different utilities for specific application scenarios.

7. Acknowledgements

This work was partially supported by a JST CREST Grant (JPMJCR18A6, VoicePersonae project), Japan, MEXT KAKENHI Grants (18H04112, 21H04906), Japan, and KDDI research, Japan.

8. References

[1] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018, pp. 1–11.

[2] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017, pp. 1–6.

[3] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior," in *Proc. ICASSP*, 2020, pp. 6699–6703.

[4] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling dnn-based speech synthesis using input codes," in *Proc. ICASSP*, 2017, pp. 4905–4909.

[5] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018, pp. 5180–5189.

[6] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling," *arXiv preprint arXiv:2010.04301*, 2020.

[7] L. Liu, J. Hu, Z. Wu, S. Yang, S. Yang, J. Jia, and H. Meng, "Controllable emphatic speech synthesis based on forward attention for expressive speech synthesis," in *Proc. SLT*. IEEE, 2021, pp. 410–414.

[8] 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. APSIPA*, 2016, pp. 1–6.

[9] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *Proc. ICASSP*, 2020, pp. 7734–7738.

[10] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 2019, pp. 5210–5219.

[11] 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.

[12] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 9, pp. 1432–1443, 2019.

[13] 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," *Proc. INTERSPEECH*, pp. 1118–1122, 2019.

[14] S. Ding and R. Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion." in *Proc. INTERSPEECH*, 2019, pp. 724–728.

[15] H.-T. Luong and J. Yamagishi, "Nautilus: a versatile voice cloning system," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2967–2981, 2020.

[16] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proc. ICLR*, 2016, pp. 1–12.

[17] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. NIPS*, 2017, pp. 1–10.

[18] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black *et al.*, "The zero resource speech challenge 2019: Tts without t," *Proc. INTERSPEECH*, pp. 1088–1092, 2019.

[19] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," in *Proc. SSW*, 2019, pp. 155–160.

[20] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé *et al.*, "Introducing the voiceprivacy initiative," in *Proc. INTERSPEECH*, 2020, pp. 1693–1697.

[21] H.-T. Luong and J. Yamagishi, "Latent linguistic embedding for cross-lingual text-to-speech and voice conversion," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 150–154.

[22] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Pretraining techniques for sequence-to-sequence voice conversion," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 745–755, 2021.

[23] H.-T. Luong and J. Yamagishi, "Multimodal speech synthesis architecture for unsupervised speaker adaptation," in *Proc. INTERSPEECH*, 2018, pp. 2494–2498.

[24] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders," in *Proc. ICASSP*, 2019, pp. 6166–6170.

[25] M. Zhang, Y. Zhou, L. Zhao, and H. Li, "Transfer learning from speech synthesis to voice conversion with non-parallel training data," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1290–1302, 2021.

[26] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. NIPS*, 2014, pp. 3581–3589.

[27] J. Williams, Y. Zhao, E. Cooper, and J. Yamagishi, "Learning disentangled phone and speaker representations in a semi-supervised vq-vae paradigm," in *Proc. ICASSP*, 2021, pp. 7053–7057.

[28] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of japanese." in *Proc. LREC*, vol. 6, 2000, pp. 1–5.

[29] 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.

[30] C. Zhang, Y. Ren, X. Tan, J. Liu, K. Zhang, T. Qin, S. Zhao, and T.-Y. Liu, "Denoispeech: Denoising text to speech with frame-level noise modeling," in *Proc. ICASSP*, 2021, pp. 7063–7067.

[31] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," in *Proc. ICLR*, 2019, pp. 1–17.

[32] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through l_0 regularization," in *Proc. ICLR*, 2018, pp. 1–13.

[33] T. V. Ho and M. Akagi, "Non-parallel voice conversion based on hierarchical latent embedding vector quantized variational autoencoder," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 140–144.