Attention-Based Speaker Embeddings for One-Shot Voice Conversion

Tatsuma Ishihara[1], *Daisuke Saito*[2,1]

[1]GREE Inc.
[2]The University of Tokyo

tatsuma.ishihara@gree.net, dsk_saito@gavo.t.u-tokyo.ac.jp

Abstract

This paper proposes a novel approach to embed speaker information to feature vectors at frame level using an attention mechanism, and its application to one-shot voice conversion. A one-shot voice conversion system is a type of voice conversion system where only one utterance from a target speaker is available for conversion. In many one-shot voice conversion systems, a speaker encoder mechanism compresses an utterance of the target speaker into a fixed-size vector for propagating speaker information. However, the obtained representation has lost temporal information related to speaker identities and it could degrade conversion quality. To alleviate this problem, we propose a novel way to embed speaker information using an attention mechanism. Instead of compressing into a fixed-size vector, our proposed speaker encoder outputs a sequence of speaker embedding vectors. The obtained sequence is selectively combined with input frames of a source speaker by an attention mechanism. Finally the obtained time varying speaker information is utilized for a decoder to generate the converted features. Objective evaluation showed that our method reduced the averaged mel-cepstrum distortion to 5.23 dB from 5.34 dB compared with the baseline system. The subjective preference test showed that our proposed system outperformed the baseline one.

Index Terms: Voice conversion, attention mechanism, speaker embedding, soft DTW, U-Net

1. Introduction

Voice conversion are techniques to modify speech signals of a source speaker to ones of a target speaker so that they sound like utterances from the target speaker. Voice conversion has a wide range of applications, such as utilization for entertainment [1], assistance of live performance [2], supports for disabilities [3], etc. Voice conversion has long histories of research. An interesting research direction is adopting data-driven approaches [4, 5, 6], in which voice conversion is treated as a machine learning problem. When adopting this approach, one of the major issues is how to collect the data which is suitable for the problem. In order to treat voice conversion as a regression problem, utilizing parallel corpora [4, 5, 6] has been widely investigated. On the other side, methods utilizing nonparallel data [7, 8, 9, 10] which treat voice conversion as a kind of reconstruction problems, have been also explored.

The function of voice conversion is divided into two functions; to ensure the consistency of the linguistic content between both the source and target speakers, and to model the speaker individuality of the target [11]. Achieving the second function of voice conversion by a small amount of nonparallel data is a challenging task, and in the case that required speech data is as small as one utterance of the target speaker, a reference utterance henceforth, it is called one-shot voice conversion. Studies for one-shot voice conversion adopt a wide variety of approaches especially to guarantee the first function, not only the second function; including pretraining and adaptation [12, 13], combination with automatic speech recognition [14, 15, 16], and content and speaker information disentanglement [17, 18], and so on. Among the various kinds of approaches for one-shot voice conversion, approaches based on an encoder-decoder framework [17, 18] are powerful and reasonable ways to realize one-shot voice conversion. For one-shot voice conversion, encoder-decoder models which possess two specific types of encoders; a content encoder and a speaker encoder, are adopted. The content encoder and the speaker encoder extract a content representation from a source utterance and a speaker representation from a reference utterance, respectively. Both the representations are fed into a decoder to construct a utterance for the target. That is to say, the two encoders are reasonable implementations of the required functions for the voice conversion.

In many voice conversion systems, it is assumed that the linguistic content is dynamic and time-varying while the speaker information is static and time-independent. Therefore the speaker representation is often modeled as a fixed-size vector. This would be a reasonable modeling strategy. However, fixed-size representation of the speaker by an utterance have two issues to be considered. First, since speech signals dynamically change in time, some parts of speaker information also would change in time. Considered the differences of mechanisms of speech production, vowels and consonants would convey different aspects of speaker information. From a viewpoint of applications, silence parts of the signals, which hardly convey speaker information, should be treated differently. Second, using a fixed-size vector as speaker representation causes a loss of information, and rich speaker information in speech would lossily compressed into a predefined capacity.

With consideration for these issues, we propose to use time-varying speaker representation for one-shot voice conversion. For extraction of the time-varying speaker information, the functions of content and speaker extractors should interlock each other. To achieve the concept, we adopt an attention mechanism for implementing time varying speaker representation, since a relation between a reference utterance and a goal of the output can be treated as a family of sequence to sequence processing. Intuitionally, this process can be interpreted as fetching speaker information from a dictionary, which is extracted from a reference utterance, depending on content. To capture an hierarchical aspects of speech, we also adopt a multi resolutional architecture similar to UNet [19].

In addition, to obtain content and speaker representation suitable for one-shot voice conversion, we design a training procedure so that the gap between training and test phases is sufficiently small. For consistent performances for conversion in the test phase, source utterances in the training phase are converted in the same way as the test phase, and converted results are com-

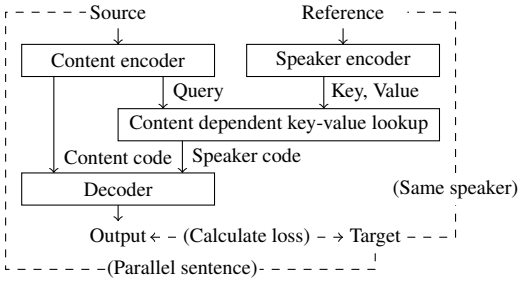

Figure 1: *Overview of our conversion model and its training procedure. The model takes two input, which are labeled source and reference. Then each input is fed into the encoder. Among output of the encoders, content code is directly fed into decoder. The rest, which are query, key and value are used for calculating time varying speaker code. Finally the codes are fed into decoder, then compared with Target, which shares speaker with reference and linguistic content with source.*

pared with the ground truth of the target. Note that this requires parallel data between a sufficiently large number of combinations of speaker pairs. However, no condition is required as to a reference utterance during the test phase. That is to say it is a method of a nonparallel approach. From another point of view, the proposed approach can be regarded as efficient utilization of the prestored parallel data. Figure 1 illustrates our method.

The rest of the paper is organized as follows. Section 2 describes an attention mechanism which is a key technique for the proposed approach. Section 3 and Section 4 describe the overview and the detailed implementation of the proposed approach, respectively. Section 5 shows the experimental evaluations and Section 6 concludes the paper.

2. Attention Mechanism

An attention mechanism [20] is a commonly-used module to model sequence to sequence processing such as machine translation. Let query $\boldsymbol{q} = \{\boldsymbol{q}(t)\}_{t=1}^{T}$, key $\boldsymbol{k} = \{\boldsymbol{k}(t')\}_{t'=1}^{T'}$ and value $\boldsymbol{v} = \{\boldsymbol{v}(t')\}_{t'=1}^{T'}$ be vector sequences, where T and T' are their sequence lengths. The attention function $\mathrm{Attention(q, k, v)}$ and its output sequence $\boldsymbol{s} = \{\boldsymbol{s}(t)\}_{t=1}^{T}$ is defined as follows,

$$\boldsymbol{s} = \mathrm{Attention}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}), \tag{1}$$

$$\begin{aligned}\boldsymbol{s}(t) &= \boldsymbol{V}\mathrm{softmax}(\hat{\boldsymbol{K}}^{\top}\hat{\boldsymbol{q}}(t)) \\ &= \frac{\sum_{t'=1}^{T'}\exp(\alpha\hat{\boldsymbol{q}}(t)^{\top}\hat{\boldsymbol{k}}(t'))\boldsymbol{v}(t')}{\sum_{t'=1}^{T'}\exp(\alpha\hat{\boldsymbol{q}}(t)^{\top}\hat{\boldsymbol{k}}(t'))},\end{aligned} \tag{2}$$

$$\hat{\boldsymbol{q}}(t) = \boldsymbol{q}(t)/\|\boldsymbol{q}(t)\|_2, \tag{3}$$

$$\hat{\boldsymbol{k}}(t') = \boldsymbol{k}(t')/\|\boldsymbol{k}(t')\|_2, \tag{4}$$

where $\hat{\boldsymbol{K}}$ and $\boldsymbol{V}$ are matrices whose t'-th column is $\hat{\boldsymbol{k}}(t')$ and $\boldsymbol{v}(t')$ respectively, and α is a hyper parameter. Note that a slightly modified version of dot product attention, in which query and key are first normalized then rescaled with α is adopted. Intuitionally, this process can be interpreted as fetching time-varying global information between source and reference sequences via $\mathrm{softmax}(\hat{\boldsymbol{K}}^{\top}\hat{\boldsymbol{q}}(t))$, when $\boldsymbol{q}$ comes from the source and $\boldsymbol{k}, \boldsymbol{v}$ come from the reference.

3. Proposed Approach and Method

3.1. Formulation of one-shot voice conversion

Let $\boldsymbol{x} = \{\boldsymbol{x}(t)\}_{t=1}^{T_x}$ be a sequence of input features, $\boldsymbol{r} = \{\boldsymbol{r}(t)\}_{t=1}^{T_r}$ be a sequence of reference features, and $\hat{\boldsymbol{x}} = \{\hat{\boldsymbol{x}}(t)\}_{t=1}^{T_x}$ be a sequence of converted features. In the following notation, bold alphabet indicates a sequence of vectors and (t) indicates the time index unless specified. Relation between those sequences are defined as follows:

$$\hat{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{r}; \theta), \tag{5}$$

where f is a conversion function parametrized by θ. Parameter optimization under a given dataset $\mathcal{X}$ can be described as:

$$\underset{\theta}{\mathrm{minimize}} \sum_{\boldsymbol{x},\boldsymbol{r},\boldsymbol{y}\in\mathcal{X}} \mathcal{L}\left(\boldsymbol{y}, f(\boldsymbol{x}, \boldsymbol{r}; \theta)\right), \tag{6}$$

where $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{x}})$ is a loss function which measures closeness between $\boldsymbol{y}$ and $\hat{\boldsymbol{x}}$. This process is performed by stochastic gradient descent or its variant such as Adam [21].

3.2. Loss functions for parallel and nonparallel settings

There are two kinds of implementations for the loss function $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{x}})$, depending on properties of the prepared training data, i.e., parallel or nonparallel settings.

For parallel training data, approaches based on dynamic time warping (DTW) is commonly used [5, 6]. While the obtained alignment is fixed in the conventional approaches, one drawback of them is the mismatch of alignment cannot be recovered during the training phase. To alleviate this, we adopt the soft DTW loss function proposed in [22], which can align output and ground truth rather than input and ground truth as in traditional DTW-based approaches. On the other hand, for nonparallel training data, the design of loss function can be straight forward, e.g., it can be frame-wise mean squared error [18, 17]. In this case $\boldsymbol{y}$ is equal to $\boldsymbol{x}$ and the $\boldsymbol{r}$ is from the same speaker as $\boldsymbol{x}$. Finally, the loss function $\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{x}})$ is defined as:

$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{x}}) = \begin{cases}\lambda_{\mathrm{MSE}}\mathcal{L}_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{x}}) & (\boldsymbol{y} = \boldsymbol{x}) \\ \lambda_{\mathrm{DTW}}\mathcal{L}_{\mathrm{DTW}}(\boldsymbol{y}, \hat{\boldsymbol{x}}; \gamma) & (otherwise)\end{cases} \tag{7}$$

$$\mathcal{L}_{\mathrm{MSE}}(\boldsymbol{y}, \hat{\boldsymbol{x}}) = \frac{1}{MT}\sum_{t=1}^{T}\|\boldsymbol{y}(t) - \hat{\boldsymbol{x}}(t)\|_2^2$$

$$\mathcal{L}_{\mathrm{DTW}}(\boldsymbol{y}, \hat{\boldsymbol{x}}; \gamma) = \frac{1}{MT}\mathrm{dtw}_{\gamma}(\boldsymbol{y}, \hat{\boldsymbol{x}}),$$

where λ_{MSE} and λ_{DTW} are hyper parameters for balancing weights, $M = \dim \hat{\boldsymbol{x}}(t)$, T is the length of sequence $\hat{\boldsymbol{x}}$.

3.3. Model architecture

3.3.1. Multi-scale autoencoder

Speech features appear in various time resolutions, which motivates us to use multi resolution architecture such as UNet [19]. Specifically, we adopt multi-scale encoders $E_{\mathrm{c}}(\boldsymbol{x})$ and $E_{\mathrm{s}}(\boldsymbol{r})$ for content and speaker information, respectively, and a multi-scale decoder D to model f. Parameters are omitted for simplicity. These encoders and decoder are related with following equations

$$\boldsymbol{w}^{(1)}, \cdots, \boldsymbol{w}^{(L)} = E_{\mathrm{c}}(\boldsymbol{x}) \tag{8}$$

$$\boldsymbol{z}^{(1)}, \cdots, \boldsymbol{z}^{(L)} = E_{\mathrm{s}}(\boldsymbol{r}) \tag{9}$$

$$\hat{\boldsymbol{x}} = D(\{\boldsymbol{w}^{(l)}\}_{l=1}^{L}, \{\boldsymbol{z}^{(l)}\}_{l=1}^{L}), \tag{10}$$

where $\{\boldsymbol{w}^{(l)}\}_{l=1}^{L}$ and $\{\boldsymbol{z}^{(l)}\}_{l=1}^{L}$ are multi-scale features extracted from $\boldsymbol{x}$ and $\boldsymbol{r}$, respectively.

3.3.2. *Attention-based speaker embeddings*

In one-shot voice conversion, speaker information is obtained from only one reference utterance. Therefore information bandwidth between the reference and output should be broad enough. To design this information path, we assume that speaker information is appeared in a content-dependent way, i.e., there are several clusters of information such as specific vowel-dependent information, specific consonant-dependent information, etc. Under this assumption, speaker information should also be conducted in a content-dependent way. For example, when synthesizing vowel, the vowel region in the reference should be regarded as more important one than the other regions such as consonant or silence parts.

Information transfer process mentioned above can be effectively modeled using an attention mechanism [20], as softmax mapping used in the attention mechanism can be interpreted as content-dependent fetching of information. Specifically, the above mentioned process is performed in our decoder in the following way:

$$\boldsymbol{c}^{(l)}, \boldsymbol{q}^{(l)} = \mathrm{split}(\boldsymbol{w}^{(l)}), \tag{11}$$

$$\boldsymbol{k}^{(l)}, \boldsymbol{v}^{(l)} = \mathrm{split}(\boldsymbol{z}^{(l)}) \tag{12}$$

$$\boldsymbol{s}^{(l)} = \mathrm{Attention}(\boldsymbol{q}^{(l)}, \boldsymbol{k}^{(l)}, \boldsymbol{v}^{(l)}) \tag{13}$$

$$\hat{\boldsymbol{x}} = \hat{D}(\boldsymbol{c}^{(1)}, \cdots, \boldsymbol{c}^{(L)}, \boldsymbol{s}^{(1)}, \cdots, \boldsymbol{s}^{(L)}), \tag{14}$$

where α is a hyper parameter, $l = (1, \ldots, L)$, and $T^{(l)}$ and $T'^{(l)}$ are the lengths of sequence $\boldsymbol{w}^{(l)}$ and $\boldsymbol{z}^{(l)}$, respectively. Intuitively, the decoder tries to reconstruct a sequence of acoustic features $\hat{\boldsymbol{x}}$ using content information $\boldsymbol{c}^{(l)}$ and content-dependent speaker information $\boldsymbol{s}^{(l)}$.

4. Details of Implementation

4.1. Acoustic feature

We used 41-dimensional mel-cepstral coefficients (MCEPs) extracted from spectral envelopes obtained using WORLD vocoder [23, 24] along with fundamental frequency contours (F_0s) and aperiodicities (APs) for acoustic features. We used 16kHz-sampled audio data, The length of short time FFT is 1024 points and 5 ms hop length. MCEPs except the 0-th coefficient were fed into neural network and converted. F_0s are converted with conventional linear regression in $\log F_0$ domain. The other features were unchanged and directly used for parameter generation.

4.2. The detail of the neural networks

We adopted UNet-like skip connections for encoders and decoder. Each block was also constructed with a short skip connection. Over all architecture can be found in Figure 2.

The input $\boldsymbol{x}$ and $\boldsymbol{r}$ were processed similarly; the sampling rate was halved $L-1$ times, then sampling rate was doubled $L-1$ times, to output $\boldsymbol{w}^{(l)}$ and $\boldsymbol{z}^{(l)}$ ($l = 1, \cdots, L$, coarse to fine order) for each sampling rate. Then the encoder's output $\boldsymbol{w}^{(l)}$ and $\boldsymbol{z}^{(l)}$ were gradually fed into decoder to calculate $\hat{\boldsymbol{x}}$.

E_{c}, E_{s} and D were constructed with fully convolutional neural networks. Weight normalization [25] was applied for each convolution kernel. Reflection padding was applied before each convolution to maintain sequence length. We adopted

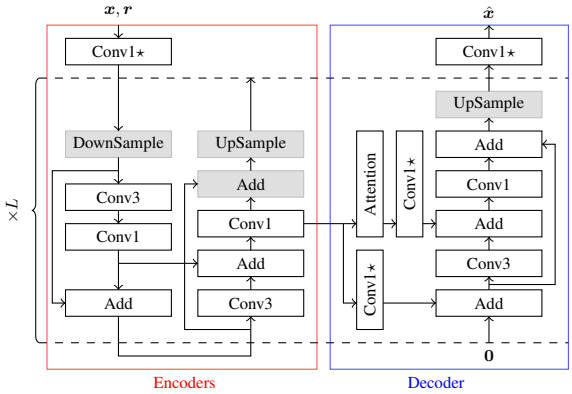

Figure 2: *Neural net architecture. Conv$\{k\}$ indicates 1D convolution with kernel size k. Each convolution layer is followed by GELU activation unless denoted with $\star$. Blocks filled with gray are not appeared in the shallowest end of repetitions. Note that there are two encoders with identical structure.*

GELU [26] for nonlinear activation function. To modify resolutions, down sampling was implemented with average pooling with stride and kernel size 2. Up sampling was implemented with nearest neighbor interpolation.

Hyper parameter settings were as follows: the number of hidden unit for each layers is 96, $\alpha = 5$, $L = 5$, $\dim \boldsymbol{q}^{(l)}(t) = \dim \boldsymbol{k}^{(l)}(t) = 16$, $\dim \boldsymbol{w}^{(l)}(t) = \dim \boldsymbol{z}^{(l)}(t) = 16 + 2^{L-l+1}$.

4.3. Training

We found that keeping distribution of $\boldsymbol{c}$ close to Gaussian distribution helps stabilizing training and generalization. Therefore we added a regularization term to the objective function to meet the above condition.

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{L}\sum_{l=1}^{L}\frac{\mathcal{L}_{\mathrm{KL}}^{(l)}}{d^{(l)}} \tag{15}$$

$$\mathcal{L}_{\mathrm{KL}}^{(l)} = \mathrm{KLD}(\mathcal{N}(\mu_{\boldsymbol{c}}^{(l)}, \Sigma_{\boldsymbol{c}}^{(l)}) \parallel \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})) \tag{16}$$

where $d^{(l)} = \dim \boldsymbol{c}^{(l)}(t)$, $\mu_{\boldsymbol{c}}^{(l)}$ and $\Sigma_{\boldsymbol{c}}^{(l)}$ are sample mean and variance of $\boldsymbol{c}^{(l)}$, respectively. The modified loss function $\hat{\mathcal{L}}(\boldsymbol{y}, \hat{\boldsymbol{x}})$ becomes

$$\hat{\mathcal{L}}(\boldsymbol{y}, \hat{\boldsymbol{x}}) = \mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{x}}) + \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}}, \tag{17}$$

where λ_{KL} is a scalar hyper parameter.

The number of parallel and nonparallel samples in batch were fixed during training, which were 16 and 32, respectively. We trained our model using Adam [21] with learning rate 10^{-4} and applied weight decay [27] with rate 10^{-4}. During training, λ_{DTW}, λ_{MSE} and λ_{KL} were 10, 1, and 0.1, respectively. Further details may be found at https://github.com/ishihara1989/ABSE.

5. Experiments

5.1. Dataset and experimental settings

We used JVS corpus [28] for evaluations. JVS dataset contains 100 Japanese parallel sentences read by 100 professional speakers. JVS also contains 30 nonparallel sentences, which are not shared between arbitrary speaker pairs. We used the first 90 speakers and the first 90 parallel sentences for training and the last 10 speakers and the last 10 sentences for evaluation. For each speaker, the longest sentence in 30 nonparallel sentences was used for a reference utterance during evaluation.

Table 1: *Mel cepstral distortions and corresponding model sizes. Smaller MCD indicates better conversion quality.*

		MCD(dB)	#params
Baseline		5.81	4.2M
Baseline	+para	5.34	4.2M
Proposed	-attention	5.28	2.0M
Proposed		**5.23**	**920k**

Table 2: *Preference score in speaker similarity and speech quality. Winner in speaker similarity and quality with statistical significance ($p < 0.05$) denoted with † and ‡, respectively. a) Baseline b) Baseline+para c) Proposed-attention d) Proposed.*

			Similarity	Quality
a)	vs	b)† ‡	$37.5 \pm 6.8\%$	$14.8 \pm 4.9\%$
a)	vs	c)† ‡	$35.0 \pm 6.7\%$	$16.0 \pm 5.2\%$
a)	vs	d)† ‡	$35.6 \pm 6.3\%$	$9.5 \pm 4.1\%$
b) ‡	vs	c)	$50.0 \pm 7.1\%$	$62.1 \pm 7.0\%$
b)	vs	d)	$46.0 \pm 7.0\%$	$53.5 \pm 7.1\%$
c)	vs	d)‡	$50.2 \pm 7.1\%$	$32.5 \pm 6.6\%$

5.2. Objective evaluation

VAE [29] based on adaptive instance normalization was used as a baseline system [30, 18]. Mel cepstrum were adopted for acoustic features instead of log mel spectrogram originally used. Since it shares component structure with our system, we could evaluate the effectiveness of each component. In addition, two extra reference systems are constructed as follows. In *Baseline+para*, a reconstruction term in the baseline's loss function was replaced with our MSE- and DTW-based one. We used the original KL divergence term instead of our $\mathcal{L}_{\mathrm{KL}}$ and gave it 0.01 weight. In *Proposed-attention*, our proposed speaker encoder was replaced with a speaker encoder which outputs fixed length speaker code. The speaker code is copied to all frames and fed into the decoder. The speaker encoder network was hand-tuned with test set to produce best objective score.

We used Mel cepstral distortions (MCD) for the objective meature [31]. The averaged MCD before conversion was 7.87 dB. Results are shown in Table 1. As it shows, Baseline+para setting greatly reduced MCD compared to Baseline which is only trained with nonparallel loss. This difference supposed to have come from that our training scheme was closer to test setting compared with Baseline. Comparing Proposed-attention with Baseline+para, as the difference between two methods is only in model architectures, suggesting our UNet based architecture effectively modeled speech dynamics, which reflected to reduction of distortion. This result showed that multi resolution finite receptive field network could be compatible with single resolution infinite receptive field network. Comparing Proposed-attention with Proposed, It suggests our time varying speaker embedding further helped reducing reconstruction error.

5.3. Subjective evaluation

We evaluated speaker similarity and sound quality with listening evaluation. Participants collected via crowd sourcing evaluated randomly assigned 10 sentence pairs converted with different methods. 5 of the sentence pairs were identical with the rest except for the order of stimuli that presented to the participants, to eliminate possibility of evaluation biases caused by ordering. As to speaker similarity evaluation, participants were presented a reference utterance before comparing two utterances converted with methods. 20 people participated for each pair of compared methods, therefore 200 utterances were evaluated for each evaluation.

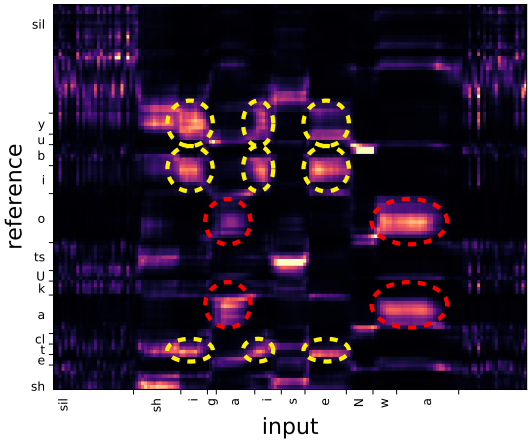

Figure 3: *Example softmax output ($l = 4$) in test set. red ellipse) Attention corresponding /a/ and /o/. yellow ellipse) Attention corresponding /i/, /e/ and /y/.*

The results are shown in Table 2. All methods using parallel objective was preferred to baseline in terms of both speaker similarity and sound quality, which was consistent with objective evaluation. As to speaker similarity there were no statistically significant difference between the other pairs. As to sound quality, Proposed-attention were less preferred to Baseline+para or Proposed. We hypothesized that this was because speaker encoder failed to embed multi resolution information into fixed sized speaker code and introduced some audible artifacts. Baseline+para, which use larger model size and infinite receptive field was compatible with our smaller and finite receptive field network in terms of both speaker similarity and sound quality. Therefore we concluded that our method can be useful for real-time conversion system without performance degradation.

5.4. Visualization of attention map

Since our model adopts an attention mechanism, visualizing attention weights can give meaningful insight. Figure 3 shows an example of attention map on test input and reference pair. It suggests that the attention map tends to make phoneme cluster, i.e., a input phoneme that belongs to a phoneme group attends to the same phoneme group appeared in reference. For example, /a/ and /o/ attends to each other, /i/, /e/ and /y/ do the same, etc. As far as we observed, these tendencies were observed at most source-target pairs, which suggests that our network learned speaker independent content representation without phoneme.

6. Conclusions

This paper has proposed a utilization of time-varying speaker representation derived from an attention mechanism and multi resolutional architecture for one-shot voice conversion. To obtain content and speaker representation suitable for the proposed approach, we have also proposed a training scheme in which the parallel data is effectively utilized. Experimental evaluations showed that the proposed approach was useful for real-time conversion system without performance degradation. In addition, We have also qualitatively analysed the attention map obtained by the proposed approach, and it suggested that our model learned linguistic-related feature without manually annotated label.

7. References

[1] Y.-J. Luo, C.-C. Hsu, K. Agres, and D. Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," *ArXiv*, vol. abs/1912.02613, 2019.

[2] R. Arakawa, S. Takamichi, and H. Saruwatari, "Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 93–98. [Online]. Available: http://dx.doi.org/10.21437/SSW.2019-17

[3] H. Doi, K. Nakamura, T. TODA, H. Saruwatari, and K. Shikano, "Enhancement of esophageal speech using statistical voice conversion," 10 2009.

[4] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 1, pp. 285–288 vol.1, 1998.

[5] 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, Nov 2007.

[6] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3893–3896, 2009.

[7] 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," *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–6, 2016.

[8] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," in *INTERSPEECH*, 2019.

[9] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[10] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: non-parallel many-to-many voice conversion using star generative adversarial networks," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 266–273, 2018.

[11] D. Saito, S. Watanabe, A. Nakamura, and N. Minematsu, "Statistical voice conversion based on noisy channel model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1784–1794, Aug 2012.

[12] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on gaussian mixture model," vol. 5, 11 2006.

[13] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Speaker adaptive training for one-to-many eigenvoice conversion based on gaussian mixture model." 01 2007, pp. 1981–1984.

[14] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5274–5278.

[15] S. Mohammadi and T. Kim, "One-shot voice conversion with disentangled representations by leveraging phonetic posteriorgrams," 09 2019, pp. 704–708.

[16] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, and H. Meng, "One-shot voice conversion with global speaker embeddings," 09 2019, pp. 669–673.

[17] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Zero-shot voice style transfer with only autoencoder loss," in *ICML*, 2019.

[18] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, and H. Meng, "One-Shot Voice Conversion with Global Speaker Embeddings," in *Proc. Interspeech 2019*, 2019, pp. 669–673. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2365

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[20] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1412–1421. [Online]. Available: https://www.aclweb.org/anthology/D15-1166

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[22] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *ICML*, 2017.

[23] 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, no. 7, pp. 1877–1884, 2016.

[24] M. Morise, "D4c, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[25] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *NIPS*, 2016.

[26] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," *ArXiv*, vol. abs/1606.08415, 2017.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[28] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "Jvs corpus: free japanese multi-speaker voice corpus," *ArXiv*, vol. abs/1908.06248, 2019.

[29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[30] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1510–1519, 2017.

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