INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Segment Level Voice Conversion with Recurrent Neural Networks

Miguel Varela Ramos[1], *Alan W. Black*[2], *Ramon Fernandez Astudillo*[1,5], *Isabel Trancoso*[1,3], *Nuno Fonseca*[4]

[1]Spoken Languages Systems Laboratory, INESC-ID-Lisboa, Portugal
[2]Language Technologies Institute, Carnegie Mellon University, USA
[3]Instituto Superior Técnico, Portugal
[4]ESTG, Polytechnic Institute of Leiria, Portugal
[5]Unbabel Lda

miguelvramos@ist.utl.pt, awb@cs.cmu.edu, ramon@astudillo.com, isabel.trancoso@inesc-id.pt, nuno.fonseca@ipleiria.pt

Abstract

Voice conversion techniques aim to modify a subject's voice characteristics in order to mimic the one's of another person. Due to the difference in utterance length between source and target speaker, state of the art voice conversion systems often rely on a frame alignment pre-processing step. This step aligns the entire utterances with algorithms such as dynamic time warping (DTW) that introduce errors, hindering system performance. In this paper we present a new technique that avoids the alignment of entire utterances at frame level, while keeping the local context during training. For this purpose, we combine an RNN model with the use of phoneme or syllable-level information, obtained from a speech recognition system. This system segments the utterances into segments which then can be grouped into overlapping windows, providing the needed context for the model to learn the temporal dependencies. We show that with this approach, notable improvements can be attained over a state of the art RNN voice conversion system on the CMU ARCTIC database. It is also worth noting that with this technique it is possible to halve the training data size and still outperform the baseline.

Index Terms: voice conversion, recurrent neural networks, deep learning, spectral mapping

1. Introduction

Voice conversion (VC) is the process of giving an utterance of a source speaker the characteristics of a target speakers voice, while keeping the original textual content. The use of VC has multiple applications such as identity switching in a text-to-speech (TTS) system, de-identification for privacy reasons, vocal restoration in cases of impaired speech, speech-to-speech translation and movie dubbing, among others.

A speaker's voice is characterized by properties such as timbre and pitch that are associated with the vocal tract and glottal source. The main goal of a voice conversion system is thus to model these physical systems. Following [1], we focus on the conversion of timbral characteristics, more specifically spectral envelope features, while leaving some prosodic characteristics of the source unaltered. Spectral features are believed to convey more speaker individuality and are easier to extract and model.

In the literature various approaches can be found that tackle the problem of VC. One of the most popular is the Joint Density Gaussian Mixture Model (JD-GMM) based technique [2, 3], which models the joint density between data of the source and target speakers using a GMM and finds local linear transformations for each Gaussian used. This approach has a tendency to both overfit the training data and produce over-smoothed converted spectra, which results in a loss of speech quality. In order to mitigate these problems, techniques such as Global Variance (GV) [4] and a mutual information criterion [5] have been proposed on top of the JD-GMM approach. Such techniques improve the results, but do not solve them completely. Besides the popular JD-GMM, techniques such a Dynamic Kernel Partial Least Squares Regression (DKPLS) [6] and Exemplar based approaches [7] have also reported some degree of success in performing voice conversion, even outperforming the JD-GMM in some cases. With the rising popularity deep learning, researchers have turned to these techniques to tackle spectral mapping for VC. Research works utilizing variations of restricted Boltzmann machines (RBM) [8], recurrent neural networks (RNNs) [9] and convolutional neural networks (CNNs) [10], have shown to be successful in performing VC. However, most of these techniques rely on training a model with aligned source and target spectral features for full utterances, without questioning the effectiveness of the alignment operation or handling the effects the high variability present within sentence. Recent work from [11] uses a phoneme level alignment to achieve a high quality alignment, although there is not any performance comparison with the traditional full utterance alignment provided.

In this paper we propose a method to reduce the variability of the input data and mitigate possible alignment mismatches that would occur more often at a full utterance level. For this purpose, we propose to break down the VC conversion process into smaller segments of syllables or phonemes, performing both alignment and conversion at this level, but incorporating additional local context to facilitate learning. We apply this approach to the high-performing RNN model proposed in [9] and obtain notable improvements over the reported performance. Most notably, we show that with this procedure it is possible to drastically reduce the amount of training data while still outperforming the original model.

The organization of this paper is as follows: Section 2 summarizes the fundamental concepts behind RNNs, which will be the basis for our model. Section 3 describes the baseline voice conversion system adopted by this paper. Section 4 describes the feature and model engineering behind our proposed VC system. Section 5 describes the experimental setup used, as well as the results obtained from our experiments, and analyses the feature alignment errors. Finally, Section 6 summarizes and presents some conclusions of this work, as well as possible fu-

http://dx.doi.org/10.21437/Interspeech.2017-1538

ture work directions.

2. Recurrent Neural Networks

2.1. Standard RNNs

An RNN is a type of neural network that is able to model sequences of variable length. For this purpose the network passes information from one time-step to another, in the same way a human being would process the meaning of each word based previous context.

In more formal terms, the standard RNN is defined in terms of an input sequence $x = (x_1, \cdots, x_T)$, the hidden state vectors $h = (h_1, \cdots, h_T)$ and the predicted output vector $y = (y_1, \cdots, y_T)$. Hidden and output values can be computed from the input by recursively applying:

$$h_t = \sigma(W_h \cdot [h_{t-1}, x_t] + b_h) \quad (1)$$

$$y_t = W_y \cdot [h_{t-1}, x_t] + b_y \quad (2)$$

where σ is a non-linear activation function and W_y, W_h and b_y, b_h are weight matrices and biases, corresponding to the network's parameters. Such a network can be trained using back propagation through time (BPTT) in a similar fashion to simpler models such as feed-forward networks.

However, the standard RNN model has a limited abilty to learn long-term dependencies and training these networks with the traditional BPTT algorithm has been proved to be extremely difficult, due to the exploding and vanishing gradient problems [12]. Motivated by this, the Long Short-Term Memory (LSTM) [13], and other variants such as the GRU [14] were proposed.

2.2. Long Short-term Memory

The LSTM was originally proposed by Hochreiter and Schmidhuber in 1997, and developed by many in following works. The current form of the LSTM is capable of learning long-term dependencies, and works very well in a wide variety of problems.

While standard RNNs have a very simple cell structure, LSTMs have a complex structures with various gates (four in total). These gates interact with each other to process information in order to better handle long-term information. Each of these gates can be viewed as separate layers that when combined together, form an LSTM cell.

The LSTM equations replace equation 1 in the RNN operation and are described as following:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{c_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t} \quad (6)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = o_t * \tanh(c_t) \quad (8)$$

where i, f, o, c, $\tilde{c}$ refer to the input gate, forget gate, output gate, cell state and shadow gate, respectively. The LSTM cell receives information continuously from the previous states via its c_{t-1} and h_{t-1} inputs. With this information, combined with the cell's parameters, the LSTM is able to decide what to read, write and forget.

2.3. Bidirectional RNNs

In order to improve the modelling of long sequences, a common technique within RNNs is the use of a bidirectional network. A bidirectional RNN consists of a group of two distinct RNNs networks that process the data from the two separate ends of the sequence. A forward RNN processes the sequence from start to end, while a backward RNN inverts the input's time axis and processes it from end to start. The outputs from both networks are then joined with a merge process. For simplicity, we admit the merge process to be the sum $h = h_f + h_b$.

3. Baseline Framework

The baseline framework adopted in this paper is based on the architecture proposed in [9], with Deep Bidirectional LSTMs (DBLSTMs). The authors proposed 6 layers of bidirectional LSTMs to map aligned Mel Generalized Cepstral (MGC) features [15], corresponding to full utterances, between source and target speakers. A set of speech parameters is extracted via STRAIGHT analysis [16]. This set of features includes a smooth spectrogram, a fundamental frequency ($F0$) trajectory and an aperiodic component, which is defined as the ratio between the lower and upper smoothed spectral envelopes in the frequency domain. The MGC features are then derived from the spectrogram. After the feature extraction process, source and target spectral features are aligned with DTW. Once the alignment is complete, the model is trained using a simple back propagation algorithm, in order to be able to map a set of source features into a set of target features. In [9], the authors keep the source's aperiodic component and the $F0$ trajectory is transformed into $\log F0$ and then converted through a popular linear conversion method, by equalizing the mean and the standard deviation of the source and target speech. Specifically, each frame is converted with the expression:

$$x' = \frac{\sigma_x}{\sigma_y}(x - \mu_x) + \mu_y \quad (9)$$

where x is the source speakers F0 value, μ_x and μ_y are the means and σ_x and σ_y are the standard deviations of the source and the target speakers data, respectively. This method modifies the global F0 level and dynamic range, while keeping the shape of the source contour [1].

An illustration of the baseline DBLSTM framework is shown in figure 1.

4. Proposed Framework: Segment Level Voice Conversion

4.1. Training Strategy

Training a big RNN model such as the one in [9] yields some of the current state of the art results. However, the model is being trained with very long sequences, which makes learning difficult, given the large variability of the data from utterance to utterance. The accuracy of the DTW alignment might also suffer from the length of these sequences.

In this work, we propose the introduction of an extra step that segments the data into several smaller pieces of data i.e. syllabes or phonemes. This will reduce the variability of the training examples seen by the model at any given time and also facilitate the DTW alignment.

Since speech data has a temporal structure deeply connected to the structure of a sentence, it is possible to break each utterance down into smaller segments such as syllables or phonemes with the help of a speech recognition system. By doing so, we are both drastically reducing the variability of each example seen by our model, as well as reducing the amount of

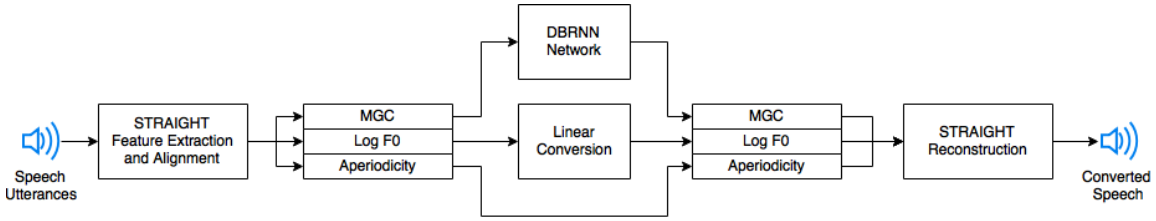

Figure 1: *Overview of the baseline voice conversion framework.*

inter segment misalignments that could occur during the feature alignment process. However, isolating single segments of syllables or phonemes, implies removing the temporal context of each training example. This implies that the RNN model will lose acess to long-term information and will not be able to capture any inter segment temporal dependencies. To tackle this problem, we propose a windowing system of triplets of syllables and phonemes to allow the model to have a peek at the previous and following context of the current segment. This windowing process is depicted in figure 2. Each segment is grouped with its previous and following segment into a triplet. In the case of the first and last segments of an utterance, the windowing is done considering the last segment of the previous utterance and the first segment of the following utterance, respectively.

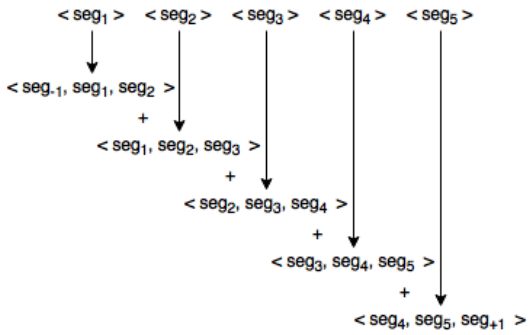

Figure 2: *Windowing technique illustration. Each segment is windowed with its previous and following segments. The last segment from the previous utterance is represented by* seg_{-1} *while the first segment from the following utterance is represented by* seg_{+1}.

4.2. Runtime Strategy

At runtime, it is still necessary to provide the additional context utilized during training. The conversion process thus applies the same windowing approach consisting in the mapping of multiple windowed segments, as many as the number of segments in the source utterance to be converted. However, for each converted segment window only the middle segment is kept, with the context converted segments being discarded. The kept segments are then concatenated and the speech is synthesized. By discarding the context segments we make sure that we are always converting with context, taking full advantage of the structure of our model.

5. Experiments

5.1. Experimental Setup

In our voice conversion experiments we use the CMU ARCTIC corpus [17]. We select two male speakers (BDL and RMS) and a female speaker (SLT) to be able to perform male to male and male to female conversions. The databases used have a total of 1132 utterances and are divided into three separate datasets, with 80% allocated for training, 20% for validation and 10% for testing. The acoustic signals are 16 bit, 16 kHz mono wave files. 49-dimensional MGC features are extracted from a real spectrogram outputted from STRAIGHT, with a 1024 Fast Fourier Transform (FFT) window size, and a 5ms frame shift. The first coefficient corresponding to the energy component of the Mel Cepstral features is removed before training, not being modelled by the network.

To extract both phoneme and syllable segments, we use the Festival Speech Synthesis System [18] which has a speech recognition module that allows to extract time locators for both phonemes and syllables. In our experiments the speech recognition module was used in forced alignment mode, where the text is provided prior to the recognition process.

The model was set up with 6 layers of bidirectional LSTMs with a 256 hidden size. The training process was conducted using the ADAM [19] optimizer and an early stopping criterion with a patience mechanism was used. The data was fed into the model in mini-batches that were padded with zeros up to the batch's maximum length, and the samples were sorted by length (bucketed) in order to reduce the amount of padding done. Furthermore, the output of each layer of the model was masked according the lengths of each sample in the batch.

The model architecture was implemented in Python with Theano [20] and training was conducted under GPU acceleration with an NVidia Tesla K40c graphical card.

5.2. Objective Evaluation

We evaluate the different architectures and compare them with each other, using the Mel cepstral distortion (MCD) as an objective evaluation metric. MCD is defined as follows:

$$MCD[dB] = \frac{10}{\ln 10}\sqrt{2\sum_{d=1}^{N}(c_d - c_d^{converted})^2} \qquad (10)$$

where c_d and $c_d^{converted}$ denote the d-th coefficient of the target and d converted Mel-cepstrum respectively. N is the dimension of Mel-cepstrum (except the energy feature) [9].

We verified that both syllable and phoneme segment level systems outperform the full utterance baseline in both male to male and male to female conversions, as shown in Table 1.

The improvement over the baseline can be explained by the reduction of the variability of the data, which limits each train-

Table 1: *Mel cepstral distortion comparison between the baseline system and the proposed segment level methods of performing voice conversion (lower is better).*

Model	Source	Target	MCD
Full Utt. ([9])	bdl (♂)	rms (♂)	6.52
Syllable	bdl (♂)	rms (♂)	5.53
Phoneme	bdl (♂)	rms (♂)	**5.38**
Full Utt. ([9])	bdl (♂)	slt (♀)	6.43
Syllable	bdl (♂)	slt (♀)	5.26
Phoneme	bdl (♂)	slt (♀)	**4.93**

ing example to a single segment and drastically increases the number of training examples, when compared to a full utterance database. With a segmented utterance database, the same segment will appear multiple times during training with variations, allowing the model to improve its learning process and generalization capabilities. This phenomenon is noticeable in the difference of performance between the voice conversion systems with syllable and phoneme based segmentations. Using a phoneme based segmentation yields a better performance over syllable segmentation, since we are further reducing the variability of the training examples and increasing the amount of data once again, while keeping the same number of utterances in our database.

In a real world scenario, it is ideal to have a system that uses as little data as possible, as parallel data for two speakers is something that is not easy to obtain. With this in mind, we tested our system against the baseline in a scenario where there is progressively fewer training data. The results on table 2 show that our system still outperforms the baseline with 50% of the data-set using a syllable segmentation strategy and 6.25% of the data-set using a phoneme segmentation strategy. This means that with as little as 49 parallel training samples, it is possible to obtain a state of the art voice conversion system.

Table 2: *Comparison of the amount of data required to achieve quality speech between the baseline and proposed solutions for a BDL(♂) to RMS(♂) conversion.*

	MCD		
Training Utt.	Full Utt. ([9])	Syllable	Phoneme
792 (100%)	6.52	5.53	**5.38**
396 (50%)	6.69	6.44	**6.14**
198 (25%)	6.88	6.98	**6.21**
99 (12.5%)	6.98	6.64	**6.28**
49 (6.25%)	7.22	6.65	**6.45**

5.3. Feature Alignment Errors

Another factor that may be contributing to the performance of our system is the additional constraints over the feature alignment process that the segmentation of the data introduces. Nevertheless, it is not immediately clear how much the alignment improves, or if it improves at all, with this additional constraint. In order be able to answer this pending question, we developed an error metric to evaluate the amount of frame mis-alignment resultant from the spectral features alignment.

In order to evaluate and compare the alignment error of both full utterance and segment level approaches, we propose an error metric that counts the number of misaligned frames with respect to its correspondent syllable. This is achieved by producing an array with the same number of frames as the MGC feature, for each source and target speaker, that in each index contains an ID number corresponding to the syllable to which that frame belongs. With both source and target arrays, we then force an alignment using the same alignment path resultant from the alignment process of both source and target MGC features. At this point, we end up with two arrays of the same size that should, if the DTW alignment was flawless, have a match of ID numbers at each index. Thus, by computing the differences between the number of IDs for the aligned array, we can get an approximate metric related to the DTW alignment error. In Table 3 we compare the percentage of misaligned frames with respect to the total number of frames in the data-set.

Table 3: *Alignment errors computed with the proposed heuristic measure for bdl to rms conversion.*

Model	Source	Target	% of alignment errors
Full Utt. ([9])	bdl (♂)	slt (♀)	13.84
Syllable	bdl (♂)	slt (♀)	**10.44**

From these empirical results, it is possible to confirm that the additional constraint on the DTW alignment process imposed by the segment level system has in fact an impact on the amount of alignment errors. We believe that the observed error reduction resultant from the segmentation does not play a major role in the increase of the system performance as much as the reduction of the variability of the training data does. However, it certainly contributes to the improvement.

6. Conclusions

We proposed an improvement of a state-of the art RNN voice conversion system that uses a speech recognition system to split the data into smaller segments, such as syllables or phonemes, but compensates for the loss of context by using a windowing approach. With an objective evaluation metric and the well known CMU ARTIC data set we verified that our system achieves state of the art results. Further experiments also revealed that, even with a very small fraction of utterances available, the proposed approach is still able to beat the baseline system. In addition to these experiments, we hypothesize over the possible sources of the observed improvement and completed the experiments by providing an analysis of the DTW alignment error. This analysis showed that the proposed approach reduces the amount of errors created by applying DTW on entire sentences. The technique here proposed opens possibilities to different directions of future work, including the integration of segment level information into the model, or the extension of the concept to a non-parallel database or even to a cross-lingual system.

7. Acknowledgements

This project was sponsored by the CMU Portugal UIP and supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 and EU project LAW-TRAIN with reference H2020-EU.3.7.653587.

8. References

[1] Z. Wu, "Spectral mapping for voice conversion," Ph.D. dissertation, Nanyang Technological University, 2015.

[2] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98, Seattle, Washington, USA, May 12-15, 1998*, 1998, pp. 285–288.

[3] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[4] H. Hwang, Y. Tsao, H. Wang, Y. Wang, and S. Chen, "Incorporating global variance in the training phase of gmm-based voice conversion," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2013, Kaohsiung, Taiwan, October 29 - November 1, 2013*, 2013, pp. 1–6.

[5] ——, "A study of mutual information for gmm-based spectral conversion," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 78–81.

[6] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *IEEE Trans. Audio, Speech & Language Processing*, vol. 20, no. 3, pp. 806–817, 2012.

[7] Z. Wu, E. Chng, and H. Li, "Exemplar-based voice conversion using joint nonnegative matrix factorization," *Multimedia Tools Appl.*, vol. 74, no. 22, pp. 9943–9958, 2015.

[8] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using speaker-dependent conditional restricted boltzmann machine," *EURASIP J. Audio, Speech and Music Processing*, vol. 2015, p. 8, 2015.

[9] L. Sun, S. Kang, K. Li, and H. M. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 4869–4873.

[10] S. Mobin and J. Bruna, "Voice conversion using convolutional neural networks," *CoRR*, vol. abs/1610.08927, 2016.

[11] N. Q. Hy, S. W. Lee, X. Tian, M. Dong, and E. S. Chng, "High quality voice conversion using prosodic and high-resolution spectral features," *Multimedia Tools Appl.*, vol. 75, no. 9, pp. 5265–5285, 2016.

[12] Y. Bengio, P. Y. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1724–1734.

[15] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis - a unified approach to speech spectral estimation," in *The 3rd International Conference on Spoken Language Processing, ICSLP 1994, Yokohama, Japan, September 18-22, 1994*, 1994.

[16] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveign, "Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based {F0} extraction: Possible role of a repetitive structure in sounds1," *Speech Communication*, vol. 27, no. 34, pp. 187 – 207, 1999.

[17] J. Kominek and A. W. Black, "The CMU arctic speech databases," in *Fifth ISCA ITRW on Speech Synthesis, Pittsburgh, PA, USA, June 14-16, 2004*, 2004, pp. 223–224.

[18] P. Taylor, A. W. Black, and R. Caley, "The architecture of the festival speech synthesis system," in *IN THE THIRD ESCA WORKSHOP IN SPEECH SYNTHESIS*, 1998, pp. 147–151.

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

[20] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.