Deep Bidirectional LSTM based RNN for Casual Speech to Clear Speech conversion

By

Shiwangi Singh

Master of Science Project

Spring 2017

UIN 677311552

Department of Computer Science

University of Illinois at Chicago

Committee Members

Dr. Forbes, Angus (Advisor)

Dr. Kenyon, Bob (Secondary member)

ACKNOWLEDGEMENTS

I would like to start by thanking my project advisor, Dr. Angus Forbes, and my project supervisor, John Novak III, for their guidance throughout the course of my Research Project. I would like to thank Electronic Visualization Laboratory and its faculty and staff for providing a great working environment, resources to conduct research, and technical support.

Secondly, I would like to thank University of Illinois at Chicago and the Department of Computer Science for giving me this opportunity to come to the United States and learn from some of the best faculty, researchers, and mentors. My special thanks to Dr. Liz Marai for introducing me to the domain of Deep Learning and sparking my interest in this field.

Special thanks to the great friends I made here at UIC: Kyle Almryde, Arthur Nishimoto, Marco Cavallo, Tarek Sawma, Mariusz Mankiewicz, Samaneh Khoshneviss, Juan Trelles and Yiji Zhang for making this journey a memorable one.

Finally, I would like to thank my family members; my mother Sarla, my father Amarjeet and my little sister Ankita for their unconditional love and support from back home in Mumbai.

TABLE OF CONTENTS

Abstract 4

1 Introduction 5

2 Recurrent neural Network architectures 6

2.1 Bidirectional RNN 7

2.2 Long Short Term Memory (LSTM) 8

2.3 Bidirectional LSTM based RNN (BLSTM) 9

3 Methodology 11

3.1 Parallel Speech Corpus 11

3.2 Phoneme alignment with Linear Time Warping 11

3.3 Feature Extraction 13

3.4 Data normalization 14

3.5 Training Bidirectional LSTM based RNN 14

3.5.1 Full sentence inputs 14

3.5.2 Zero padding and masking 14

3.5.3 Backpropagation Through Time 14

3.5.4 Prediction 15

3.5.5 Experiment 15

3.5.6 Conversion of RNN output into speech 16

4 Results 17

5 Conclusion 22

6 Future work 23

7 References 24

ABSTRACT

Clear Speech consists of crisp, loud, and slowly spoken sentence(s) for increased intelligibility where the utterances are distinguishable and audible without much strain on the human ear. For individuals with auditory comprehension disabilities, clear speech would be easier to comprehend than casual or normal speech. In this effort, we try to automate the conversion of casual speech to clear speech using machine translation.

In the current work we develop a Recurrent Neural Network (RNN) that uses parallel corpora with one set of corpus as input and predicts output by the parameters learnt from another corpus for this translation. Basing our study from literature, we make use of Deep Bidirectional Long Short term memory (LSTM) based RNN model because of its proven effectiveness in speech translation domain. To generalize, we convert voice A to voice B by passing voice A through multiple forward-backward LSTM exploiting its capability to learn voice B from both past and future time steps in voice A trained to regress voice B via backpropagation through time.

We performed our preliminary experiments on a male to female speech conversion task, following which we were able to extend the developed pipeline for casual to clear speech conversion.

1 INTRODUCTION

Speech Conversion is the process of modifying the audio signals in order to transform one speech form into another. There are many uses for speech conversion, such as translating the speech of one language to another language, voice conversion from a male speaker to a female speaker, Text-To-Speech (TTS) conversion, generating new voices and even for entertainment purposes. The task of speech conversion is not trivial. There is a broad literature in both Audio Processing domain and Statistical or Machine learning domains. We will be focusing on the latter.

In the early days, Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) were used to deal with temporal aspect in Speech recognition [1]. Kain et al. used GMM for spectral voice conversion for TTS [2]. Toda et al., used GMM of the joint probability of source and target for spectral conversion [3]. Desai et al., used an Artificial Neural Network (ANN) for voice conversion and performed a comparative study between ANN and GMM [4].

With the emergence of Deep Learning techniques in recent years, much progress has been made in Acoustic Modeling [5,6]. Chen et al. used Deep Neural Network (DNN) [7] for voice conversion and Nakashika et al. used Deep Belief Network (DBN) to learn higher order features for source and target speaker [8]. The problem with DNNs and DBNs was that they were not able to capture temporal aspect of the speech spectrum.

Recurrent Neural Networks (RNNs) overcomes this problem by making use of recurrent connections in the Neural Network that helps retain information over a long period of time [9]. Two decades ago this would have been a problem, because training RNNs were computationally expensive but thanks to today's modern infrastructure and computing power, this is no longer a limitation. Despite having enough resources, there were still difficulties in training RNNs with long-range dependencies [10]. Additionally, they faced a problem of exploding and vanishing gradients [11].

Long Short Term Memory (LSTM) [12] could memorize or forget long-range contextual information in sequences by storing temporal information in a memory block. Bidirectional LSTM based RNNs further extends its capabilities in learning long-range contextual information in both past and future directions [13,14]. Some of the previous work done using Bidirectional LSTM are in speech recognition [15], TTS [16], feature enhancement [17] and so on.

We based our initial study on the work done by Sun et al. [18] in Voice conversion using Deep Bidirectional LSTM based RNN but we apply this technique for Casual to Clear Speech conversion.

2 Recurrent neural Network architectures

A typical RNN [19] is used for sequential modeling by having hidden-to-hidden connections as opposed to a feedforward artificial neural network [20]. These hidden-to-hidden connections allow the neurons in the current hidden layer to use the contextual information's 'states' from the previous time steps as well as from the previous hidden layer. A representational example of a typical RNN is shown in figure 1 below.

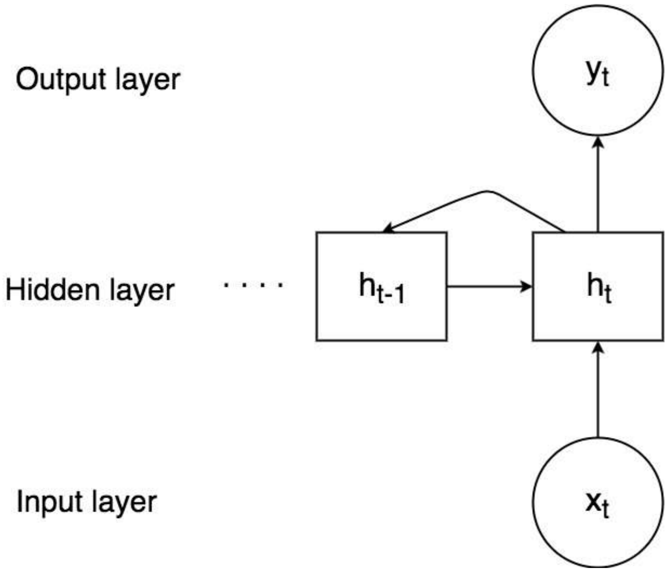


Figure 1 The basic structure of a typical RNN

We can formalize the interaction of hidden-states and output of the RNN by the following set of equations.

If $x_t = [x_1, x_2, x_3 \dots\ x_T]$ are the input data sequence, $h_t = [h_1, h_2, h_3 \dots\ h_T]$ are hidden states and $y_t = [y_1, y_2, y_3 \dots\ y_T]$ are the output data sequence, where $t = 1\ to\ T$, the equations below describe how each time step in the sequence is calculated:

$$h_t = \sigma\,(\,x_t\,.\ W_{xh}\ +\ h_{t-1}\,.\ W_{hh}\ +\ b_h\,)$$

$$y_t = h_t\,.W_{hy} + b_y$$

where W_{xh} represents input-to-hidden weights, W_{hh} represents hidden-to-hidden weights, W_{hy} represents hidden-to-output weights, b_h is hidden biases and b_y is output biases.

A RNN can be unfolded in time and can be seen as time steps $[t-1,\ t,\ t+1]$ as shown in figure 2. Each node in this figure represents a layer of network units at a single time step and the same weights are reused every single time step. That is to say, W_{xh} will be reused for every single time step for the input-to-hidden connection and similarly W_{hh} will be reused for every single time step for the hidden-to-hidden connection.

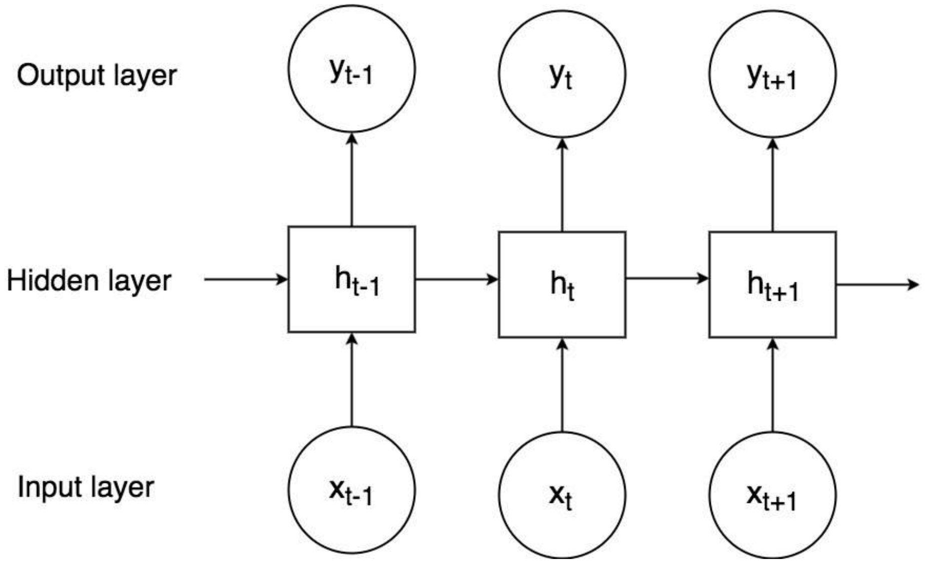


Figure 2 Unfolded RNN

2.1 BIDIRECTIONAL RNN

A Bidirectional RNN [21] (see figure 3) makes use of both the past and the future contextual information for every time step in the input sequence in order to calculate the output sequences. It has two separate hidden layers with forward states and backward states which computes the contextual information by iterating through $t = 1 \; to \; T$ in the forward or positive direction and $t = T \; to \; 1$ in the backward or negative direction. Backward states are analogous to reversing the time steps in the forward states. The forward and backward hidden states are computed and merged using the following equations:

$$h_t^f = \sigma\left(x_t \,.\, W_{xh^f} + h_{t-1}^f \,.\, W_{h^f h^f} + b_{h^f} \right)$$

$$h_t^b = \sigma\left(x_t \,.\, W_{xh^b} + h_{t+1}^b \,.\, W_{h^b h^b} + b_{h^b} \right)$$

$$y_t = h_t^f \,.W_{h^f y} + h_t^b \,.\, W_{h^b y} + b_y$$

where h_t^f and h_t^b represents the forward states and backward of the hidden layer with connections in forward and backward directions respectively. $W_{\_h^f}$ and $W_{\_h^b}$ represents weights for the forward and backward directions respectively. Output y_t is obtained by summing the parameterized hidden states for every time step.

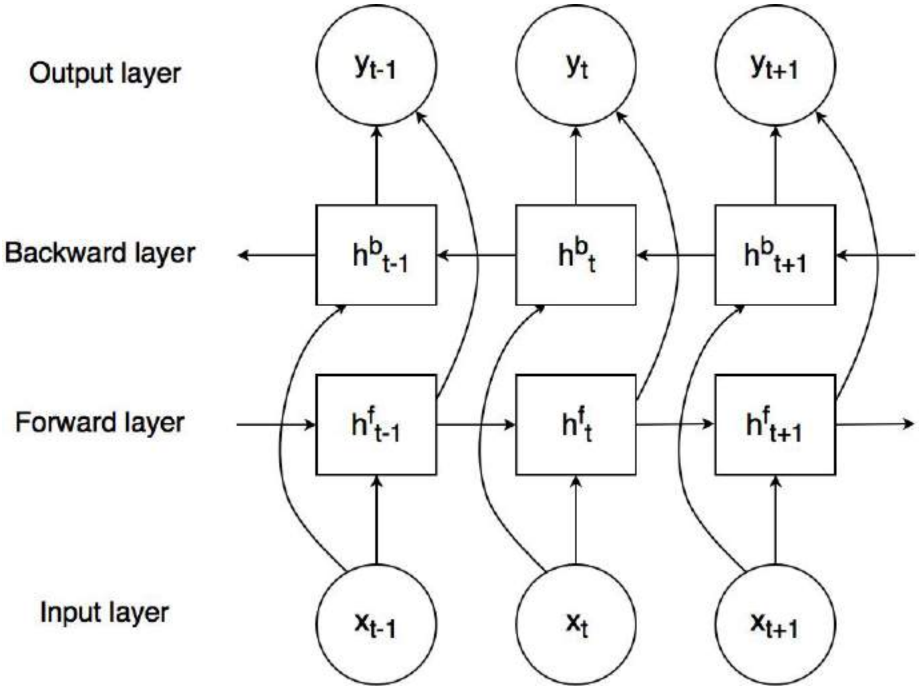


Figure 3 A Bidirectional RNN

2.2 LONG SHORT TERM MEMORY (LSTM)

Long Short Term Memory blocks were introduced by Hochreiter and Schmidhuber [12], which allowed the recurrent neural networks to capture long-range contexts in a sequence. The LSTM blocks also overcame the vanishing gradient [10] problems in the Recurrent Neural Network training. These memory blocks are similar to a computer's memory where information can be read from, written to, and stored or reset. LSTM block consists of a self-connected cell and three gates that decide when to perform write, read, and reset operations using input gate, output gate, and forget gate respectively. Figure 4 shows a LSTM block. Cells are responsible for the learning process, back-propagating error through time and updating weights during training. The hidden layer activations in a memory unit are calculated using the following equations:

$$i_t = \sigma\,(W_{xi} \,.\, x_t \;+\; W_{hi} \,.\, h_{t-1} + \; W_{ci} \,.\, c_{t-1} \;+\; b_i)$$

$$f_t = \sigma\,(W_{xf} \,.\, x_t \;+\; W_{hf} \,.\, h_{t-1} + \; W_{cf} \,.\, c_{t-1} \;+\; b_f)$$

$$c_t = f_t \,.\, c_{t-1} + \; i_t \,.\, \tanh(W_{xc} \,.\, x_t \;+\; W_{hc} \,.\, h_{t-1} \;+\; b_c)$$

$$o_t = \sigma\,(W_{xo} \,.\, x_t \;+\; W_{ho} \,.\, h_{t-1} + \; W_{co} \,.\, c_t \;+\; b_o)$$

$$h_t = o_t \,.\, \tanh\;(c_t)$$

where i_t, f_t and o_t represents input gate, forget gate and output gate respectively and c_t are the cell states. All the gates have sigmoid activations and cells have $tanh$ activations. In the LSTM block diagram, the gates are multiplicative but the cell basically sums the new input state with its previous state.

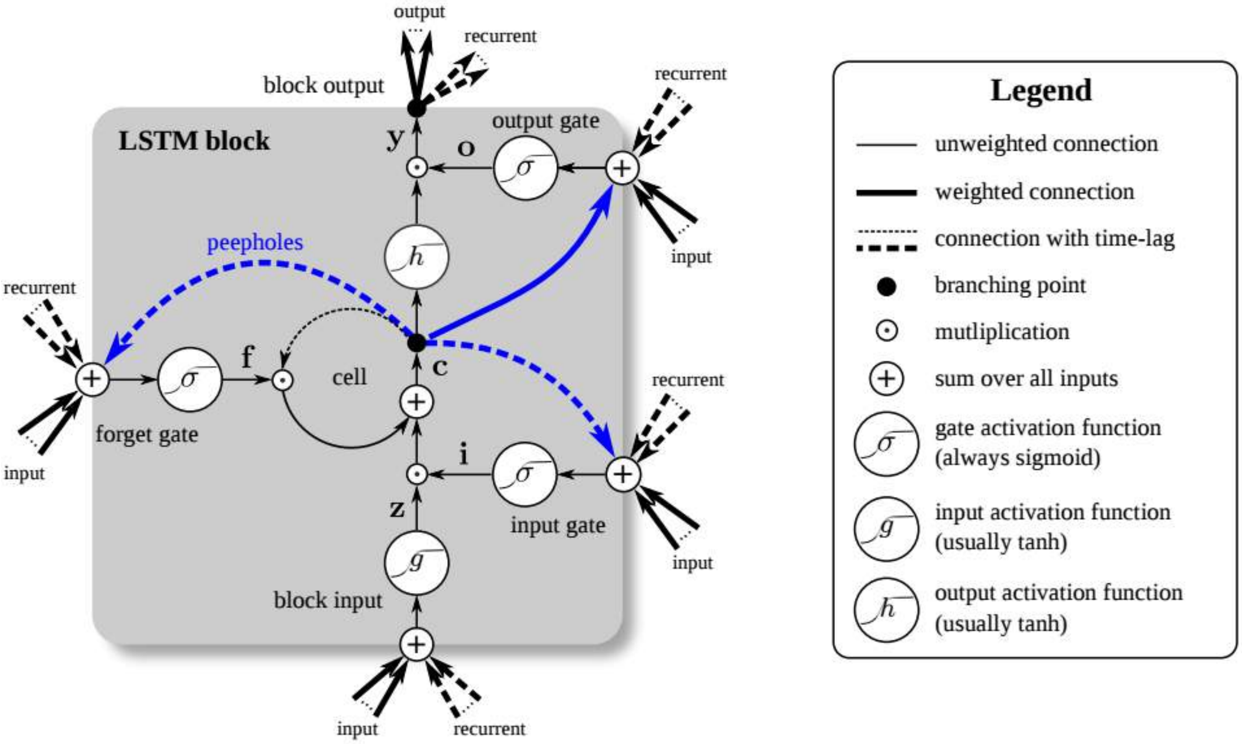


Figure 4 LSTM Block [12]

2.3 Bidirectional LSTM based RNN (BLSTM)

Combining the advantages of both Bidirectional RNN and LSTM, we can exploit the long-range context dependencies in the past as well as the future time steps while performing sequential modeling. Figure 5 shows the unfolded BLSTM model for three time steps $t-1, t, t+1$.

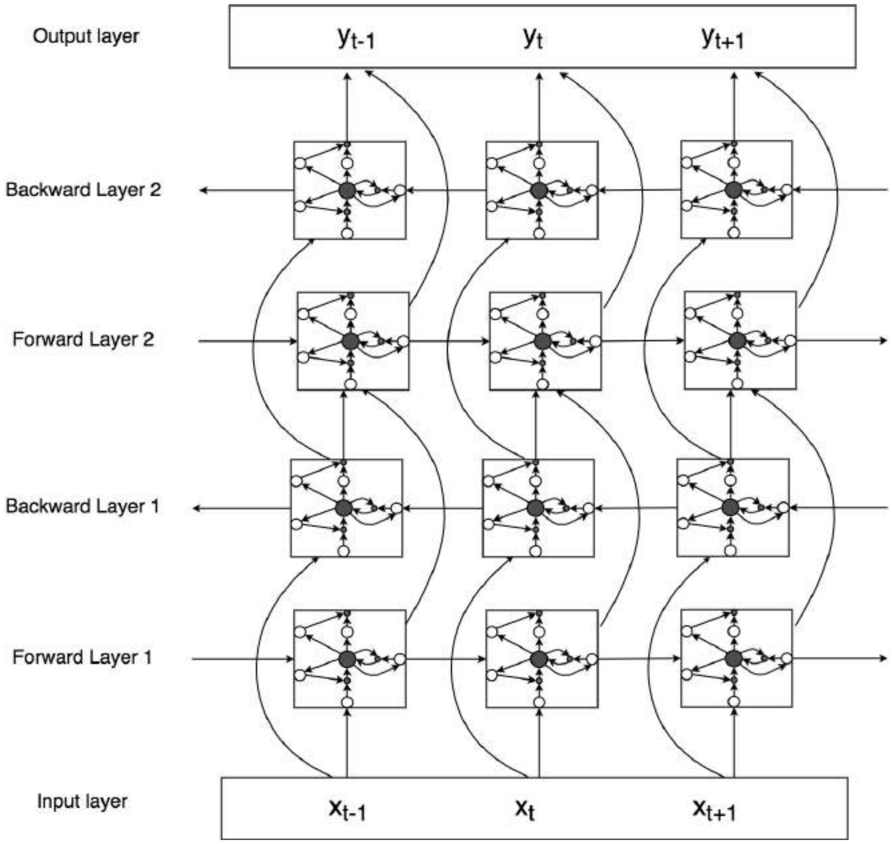


Figure 5 unfolded BLSTM based RNN model

The input is fed into both the forward and the backward LSTM layer. The output layer or the next hidden layer (consisting of forward and backward LSTM layers) receives an input by joining the forward and the backward LSTM layers. There are no hidden-to-hidden connections between forward and backward layers. The equation for the output y_t is similar to that of Bidirectional RNN. We developed BLSTM for speech conversion and from here on, we will be referring to this particular model in our established pipeline.

3 METHODOLOGY

The approach to converting Casual speech to Clear speech involved a multi-process pipeline which will be discussed in detail below. Figure 6 displays the entire processing pipeline.

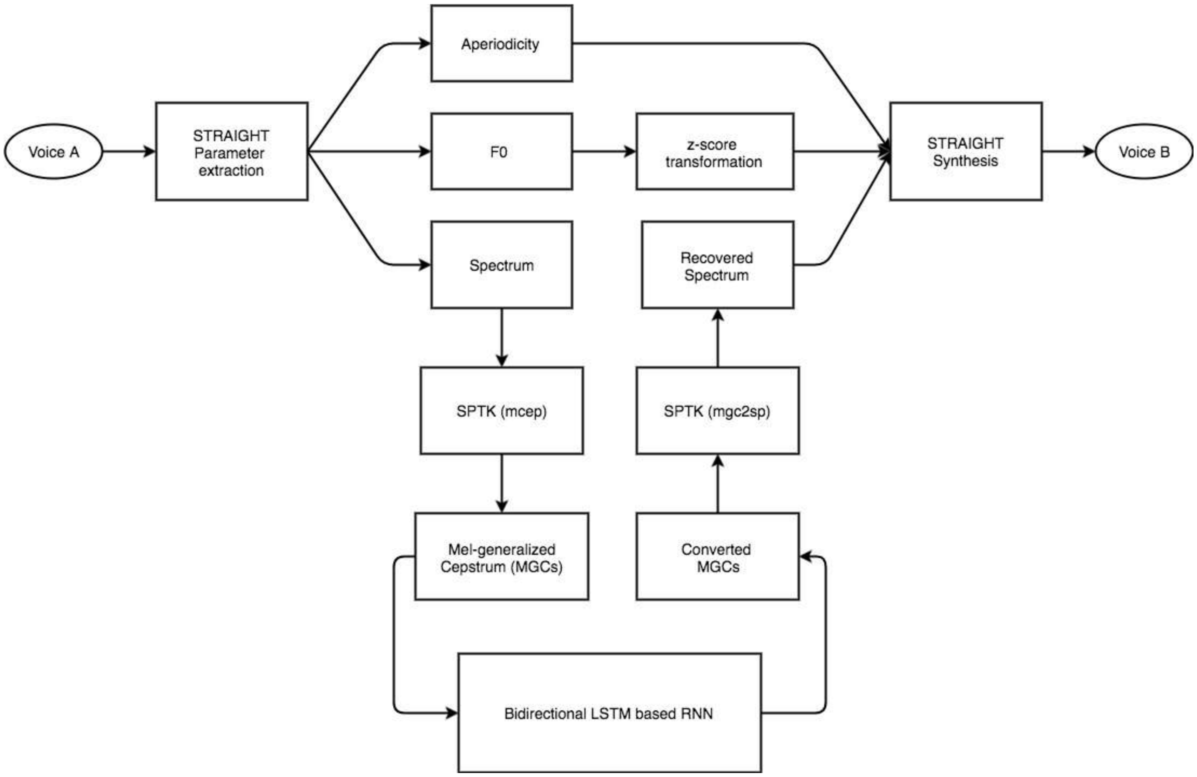


Figure 6 The system architecture

3.1 PARALLEL SPEECH CORPUS

A parallel speech corpus is a collection of pairs of spoken sentences in which each sentence in the pair has the same content but is spoken in different ways, either by the same speaker or a different speaker. For our training, we are using two corpora; The first is from the CMU ARCTIC Databases [22] and the other from the Basic English Lexicon (BEL) corpus *[23].* From the ARCTIC Corpus, we used the Scottish English male speaker (AWB) and an US English female speaker (SLT). The second corpus consists of parallel sentences spoken by the same female speaker using two different styles of speech, defined as *Speech A* and *Speech A' respectively*. *Speech A* is a normally spoken sentence, which we defined as casual or normal speech. *Speech A'* is a slow, clearly and/or intelligibly spoken sentence, which we defined as clear speech. Our initial experiments were focused on converting male to female speech from the ARCTIC corpus but this model can be easily extended to a Casual-to-Clear speech conversion model.

3.2 PHONEME ALIGNMENT WITH LINEAR TIME WARPING

In the parallel corpus, each pair of sentences has different lengths. The start and end time of the two sentences could be different, and in most cases, are different. This difference can be seen at the sentence level, word level or phonemic level. In figure 7, the sentence “Will we ever forget it?” is spoken differently in the two corresponding parallel utterances. You could clearly see, the male and the female speech differ in sentence length. Both in terms of the start and end times at word level as well as the phonemic level. Our target was to make the two corresponding sentences align at the phonemic level.

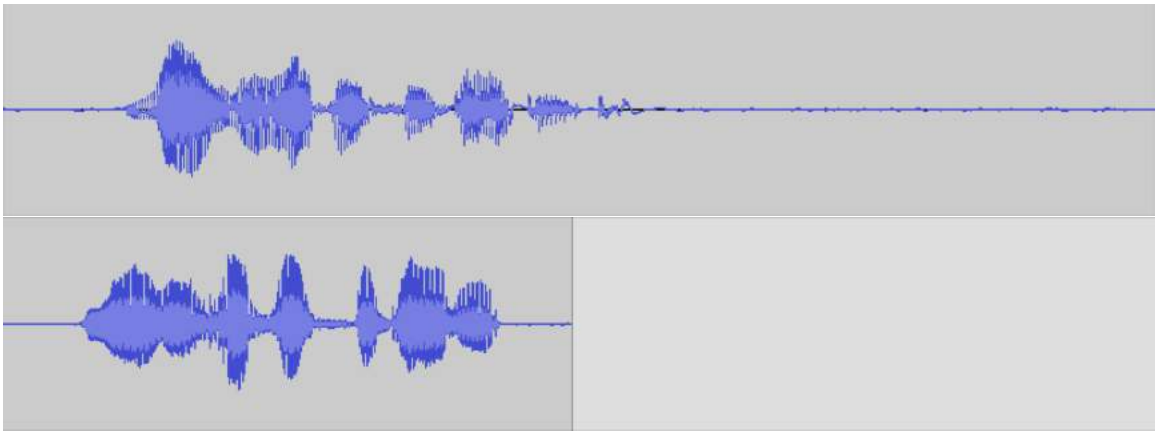


Figure 7 Waveform of male (top) and female (bottom) speakers speaking the phrase, "Will we ever forget it?"

To accomplish this, we perform linear time compression on the auditory waveform of the male speaker so as to align to the female speaker. After the alignment is done, we should be able to get a linear relationship across the time steps of the two waveforms. The Python programming language was used to extract Phoneme ‘anchor points’, which were the end times of phonemes identified in the speech. These anchor points and their parent sound files were then processed using the TSM toolbox [24] for MATLAB[1], which uses the WSOLA time-scale modification algorithm [25] to perform the actual phoneme-alignment and time-warping. Figure 8 shows the results of this transformation process.

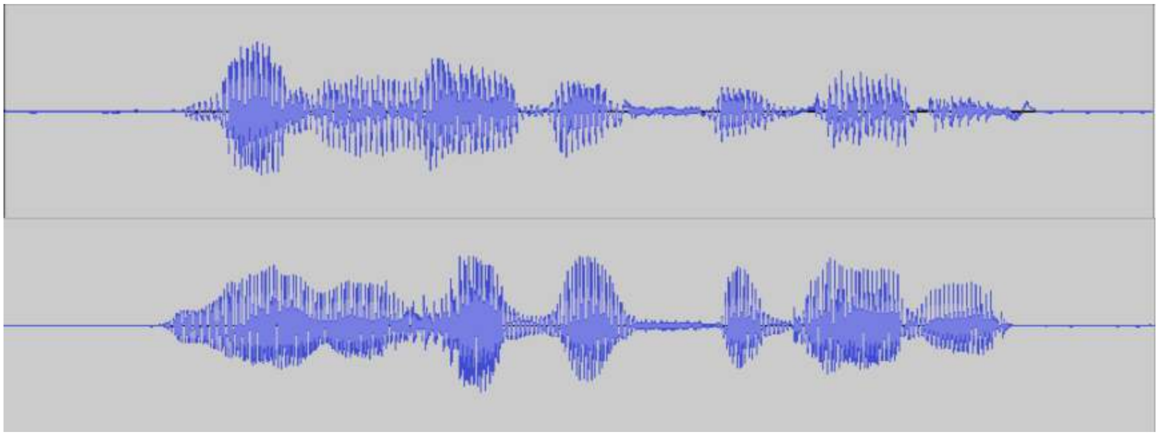


Figure 8 Waveform of “Will we ever forget it” by male (top) and female (bottom) speakers after alignment

[1] https://www.mathworks.com/

[2] For a detailed review, see [26]

3.3 Feature Extraction

Feature extraction in this context is the process in which we extract a compact representation of speech that holds good statistical properties and makes it feasible for training a RNN. Tandem-STRAIGHT [26] is used to extract what we call STRAIGHT parameters. When a sound file is given as an input to Tandem-STRAIGHT, it breaks down the waveform into three parameters, namely Aperiodicity, F0 and residual Spectrum. The parameter that goes as an input to the RNN is derived from the residual Spectrum. The residual Spectrum has a dimension of approximately 500 spectral components and a sequence length equal to the number of frames of a given speech sequence. This results in an input that is far too dense to be supplied to the neural network which would otherwise yield impossibly long training times. Therefore, we take this residual Spectrum and extract Mel-generalized Cepstral (MGCs) Coefficients[2] using SPTK [28].

MGCs are well suited as input to our model, due to the fact that:

a) They are a very concise representation of spectral waveform, compressing the 500 Spectral components into 50 Coefficients.
b) They operate on Mel-frequency scale which is a very good approximation of how a human ear perceives sound.
c) They capture very good statistical properties which makes it ideal for machine learning. MGCs take the form of 50 Coefficients corresponding to each time step or number of frames in the sequence.

As suggested in [18], we withhold the first energy component from these 50 MGC Coefficients which left us with 49 dimensional coefficients for each frame in the sequence. Thus, a *49 x Number of frames* matrix is used as inputs to our RNN. We extracted the MGCs for both source and target speech independently.

This process of extracting the MGCs from the residual Spectrum is reversed following the output MGCs from our model. As seen in figure 6**,** we use the converted MGCs to recover the spectrum using the SPTK's mgc2sp method. This recovered spectrum is used to synthesize waveform using STRAIGHT synthesis. The other two STRAIGHT parameters are handled a bit differently. We do not change the aperiodicity as it does not make a significant difference for source to target speech conversion and therefore we pass it directly to STRAIGHT. Source F0 is linearly converted to target F0 by using z-score transformation which is obtained using the following equation [29]:

$$log\ fo_{converted} = \mu_{target} + \frac{\sigma_{target}}{\sigma_{source}}(\ logfo_{source}\ - \mu_{source}\ )$$

where sigma(source) and sigma(target) are the standard deviations of the source and the target F0 respectively, mu(source) and mu(target) are the means of the source and target F0 respectively. This conversion will give you the pitch of the target speech. Thus, recovered spectrum, unchanged aperiodicity and z-score transformed F0 are used to synthesize the converted speech or waveform.

[2] For a detailed review, see [26]

3.4 DATA NORMALIZATION

We normalize the inputs, the 49-dimensional Mel-cepstrals, so as to make the distribution have a zero mean and unit variance. We perform this operation by simply by subtracting the mean from each sample and dividing by standard deviation across the 49-dimensional coefficient axis.

3.5 TRAINING BIDIRECTIONAL LSTM BASED RNN

3.5.1 *FULL SENTENCE INPUTS*

We train the Bidirectional LSTM based RNN on one complete sentence at a time making use of full contextual information from both the past and the future time steps, or more precisely, using both the forward states and the backward states in the hidden sequence. Corresponding to each input sequence, we obtain a converted output sequence. We will talk about this in details in the following sections.

3.5.2 *ZERO PADDING AND MASKING*

The input shape takes the shape of tensor with dimensions *(number of samples, sequence length, number of features).* The number of features remains the same for each sequence since we have a 49-dimensional input sequence. But input sequences are not of the same length, that is to say they vary in sequence lengths. Our model takes a fixed tensor as inputs, so we added zero padding and masking to the RNN inputs. Zero padding is simply adding matrix of zeroes at the end of the sequence to match the maximum length sequence in the training samples. Masking is a matrix of 1s and 0s that tell the RNN what the actual sequence length was. This is done by adding 1s for the actual sequence length and 0s for the extra padding.

3.5.3 *BACKPROPAGATION THROUGH TIME*

The training in a bidirectional LSTM based RNN is similar to a typical RNN because there are no connections between the forward and the backward hidden layers. We can therefore train the forward hidden layer just like a regular RNN and we can train the backward layer by reversing the direction of the training loop. We then join the hidden states for the particular hidden layer giving us the output h_t for that layer. Below is the training loop in Bidirectional LSTM based RNN (reproduced from [21]):

```
1. Forward Pass
        for t = 1 to T:
                Forward pass for the forward state storing the activations at every time
                step
        for t = T to 1:
                Forward pass for the backward state storing the activations at every
                time step
        for all t:
                Forward pass for all the predicted outputs using activations from both
                the forward and backward layers (this will the output of the current
                hidden layer)
```

2. **Backward Propagation Through Time**

```
for all t:
    Backward pass for the output layers, calculating the derivatives for each
    time step (the output layer can be the succeeding hidden layers)
for t = T to 1:
    Backward pass for the forward states using the calculated derivatives
    from the output layers
for t = 1 to T:
    Backward pass for the backward states using the calculated derivatives
    from the output layers
```

3. **Update Weights**

3.5.4 Prediction

After the training is complete, we query the model with source speech or in this case source MGCs, and we want to be able to predict the converted MGCs. The equation below gives the output of the Bidirectional RNN and the output is calculated by [30]:

$$P\left(\frac{y_t}{x_{d\ (d \neq t)}}\right) = \sigma\ (W_y^f\ .h_t^f\ +\ W_y^b\ .h_t^b\ +\ b_y)$$

We can apply a similar equation to Bidirectional RNN with LSTM memory blocks as well. Thus for predicting the converted sequence at each time step, we use the sum of the parameterized forward and backward states taking advantage of the long-range past and future contexts which comes from the LSTM block output.

3.5.5 Experiment

We performed a supervised learning algorithm for this conversion where the input to our model was a male speaker and the target was a female speaker for the first corpus. Similarly, for the second corpus, the input was casual speech and the target was clear speech.

Our network architecture was [49, 128, 256, 256, 128, 49] where each value represents a hidden dimension for bidirectional LSTM layers. Each hidden layer has a forward and a backward LSTM layer with those many hidden units. Our objective was to minimize the L2 norm loss between the prediction and the targets. We used Adam optimizer [31] as our update function with a learning rate of 1e-3 and trained the model for 1000 epochs. A **NVIDIA Quadro M4000 GPU** was used for training. The model developed and training performed was using the deep learning libraries theano [32] and theano based lasagne [33].

3.5.6 CONVERSION OF RNN OUTPUT INTO SPEECH

Once we obtain the converted MGCs of the target speech, we can now reverse the process of feature extraction in order to get the converted waveform or speech. First and foremost, we need to denormalize the MGCs which can be done by using the female data statistics in terms of mean and standard deviation. The following equation will denormalize the MGCs:

$$mgc_{denormalized} = \mu_{target\_mgc} + \sigma_{target\_mgc} * mgc_{output}$$

Initially before passing the MGCs to our model, we withheld the first coefficient vector which is the energy component from 50-dimensional MGC Coefficients. We need to add the energy component back again in order to recover a spectrum from the denormalized MGCs. This recovered spectrum is then fed into STRAIGHT synthesizer along with linearly converted F0 and aperiodicity which gives the synthesized waveform. This conversion process can be seen in Figure 6.

4 Results

The following section shows the results obtained from our developed speech conversion system. The results will focus on the output of the Bidirectional LSTM based RNN model in order to convert one speech into another. We will discuss the results obtained from both male to female conversion and Casual to Clear Speech conversion. Although the results achieved were on an overfitted model, with longer training and hyperparameter optimization, we can achieve desired results on validation data with our developed system. We can divide our results into three main stages: Converted MGCs, Recovered Spectrum and Synthesized Speech.

I. Converted MGCs

This is perhaps the most important result obtained from our Recurrent Neural Network part of the system. Using the parameters learnt by the model, converted MGCs are the predicted sequences when we query the model with a source speech. Figures 9 and 10 show a comparison of MGCs obtained from source, target and model output of casual to clear speech conversion. Converted MGCs take a shape of *(MGC Coefficients = 50, sequence length).*

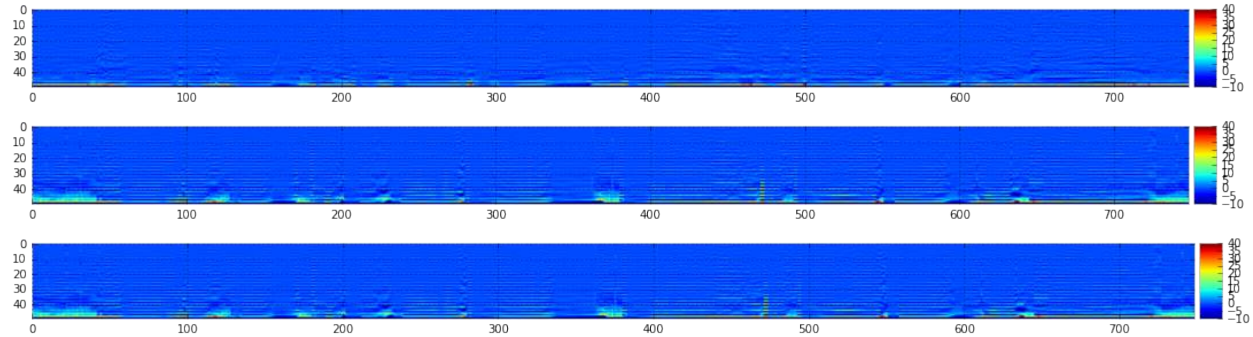
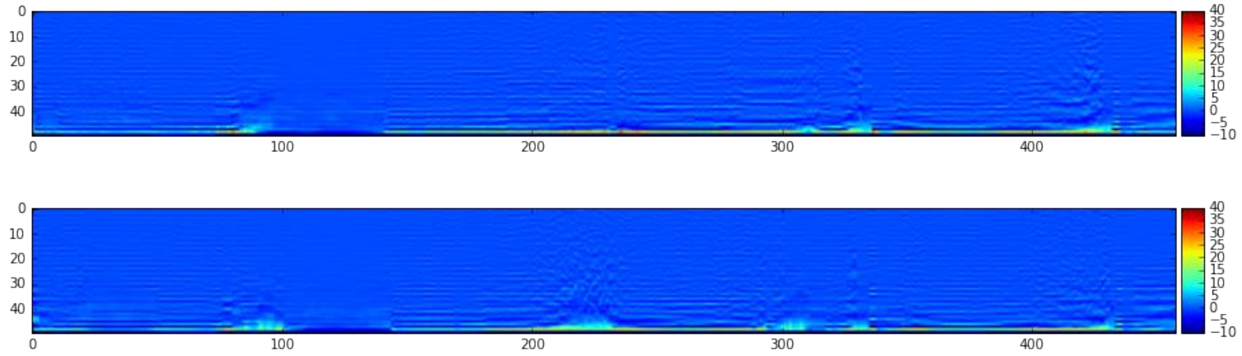


Figure 9 Male (top), Female(middle), and Output (bottom) MGCs of the ARCTIC corpus



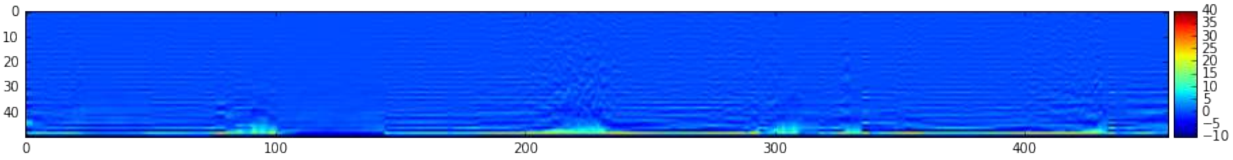


Figure 10 Causal (top), Clear (middle), Output (bottom) MGCs of the Causal-to-Clear speech corpus

Although the MGCs are not known to have very meaningful structures in them, we were surprised to see high energy bands (also referred to as harmonics) which showed matching shapes across target and output MGCs, acting as an informal validation technique.

II. Recovered Spectrum

Figures 11 and 12 shows the recovered spectrum of source, target and model output spectra. From our observations, recovered spectrum was very sensitive to the quality of the converted MGCs. If the converted MGCs did not closely match the target MGCs or there the model was trained enough, we observed some frames with abysmal values which affected the overall synthesis of Speech. For the waveform to be of good quality, our recovered spectrum needs to be well constructed. Thus, this makes the recovered spectrum to be the metric of utmost importance after learning curve and regression loss to give us an indication of how well our model is performing. Recovered Spectrum takes a shape of *(Spectral Component = 513, sequence length).*

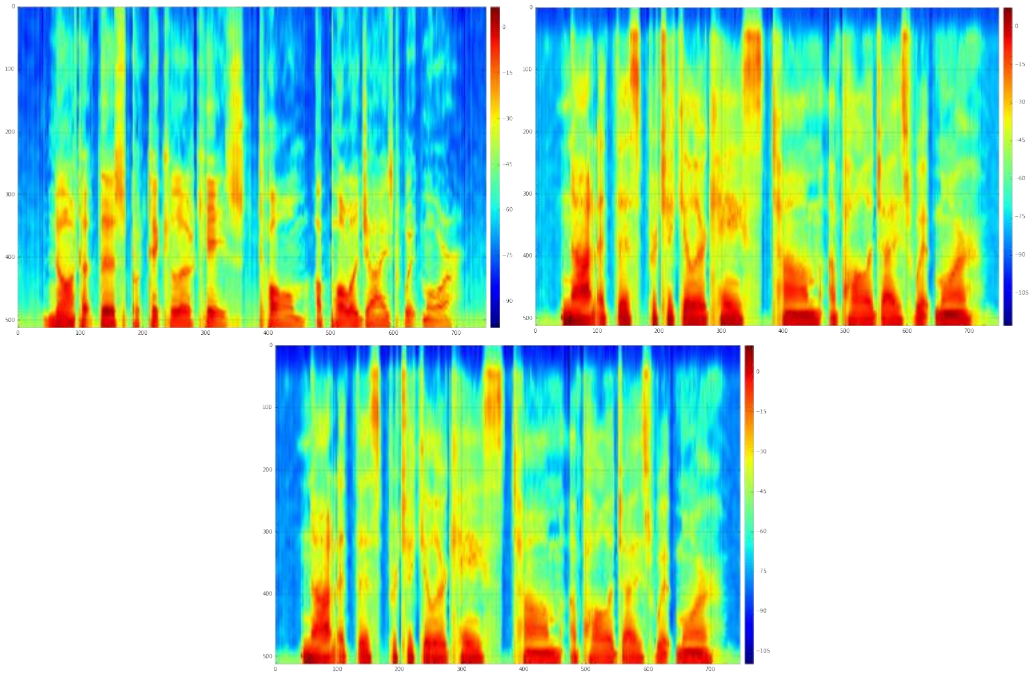


Figure 11 Male (top left), Female (top right), and Output (bottom) recovered spectrums from the ARCTIC corpus

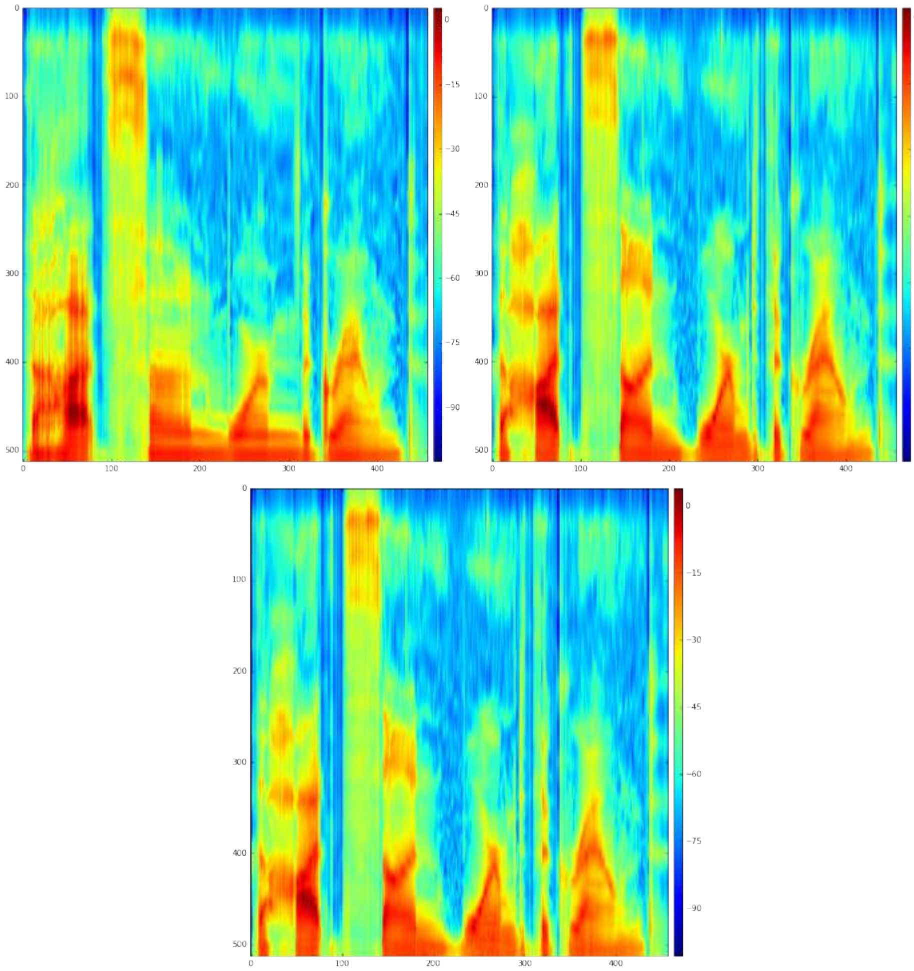


Figure 12 Casual (top-left), Clear (top-right), Output (bottom) recovered spectrums of the Casual-to-Clear speech corpus

Recovered Spectrums can be read from the bottom up: the red areas are high energy values and the blue areas are low energy values. Red areas indicate that there are spoken words or utterances at those time steps in the sequence. If we closely observe the top-right and the bottom Spectrums, we see a striking similarity between the two, but we can also see some differences as well. For instance, in the male-to-female conversion, the output spectrum does not have dark red values. This is an ideal output, in practice without having an overfitted model, we should see some deteriorations in the Output recovered spectrum.

III. Synthesized Speech

This is the final step where we synthesize speech waveform from the recovered spectrum and can listen to the sound. This section is kept only to show visually how the produced speech looks like. Figures 13 and 14 shows the spectrogram of the source, target and converted speech.

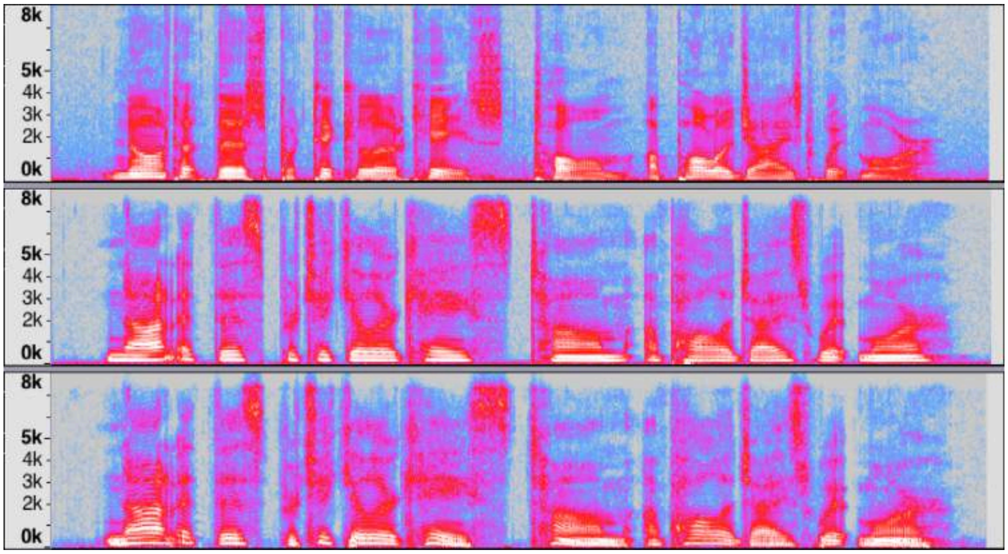


Figure 13 Male (top), Female (middle), Output (bottom) of the synthesized speech of the ARCTIC corpus

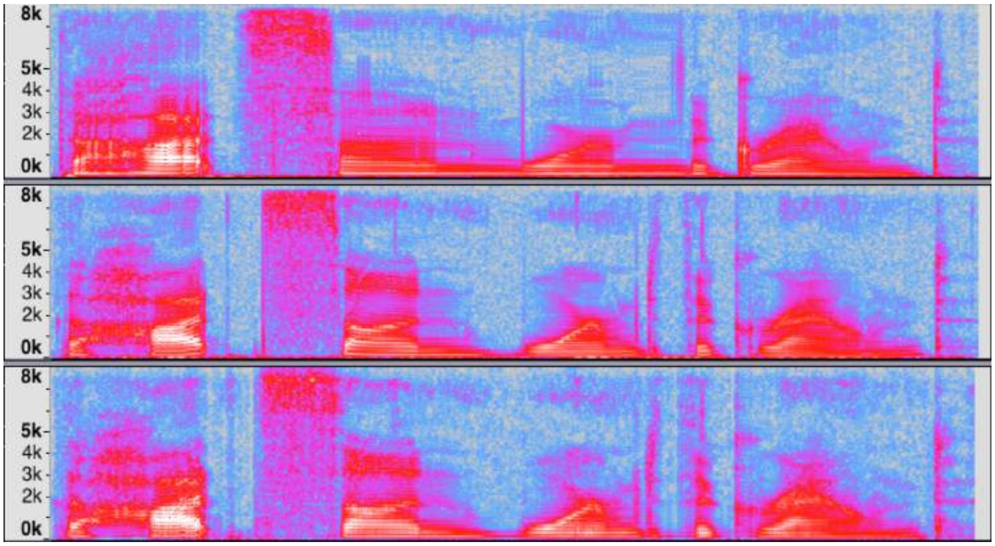


Figure 14 Male (top), Female (middle), Output (bottom) of the synthesized speech of the Clear-to-Casual speech corpus

In the male-to-female conversion, when we heard the Output sound, we observed two things: First, it closely resembled the female voice and second, it sounded a bit mechanical as compared to the female target speech. In the very first utterance, the high energy bands are upward facing in the case of the female target spectrum(middle) and facing downwards in the case of the Output spectrum(bottom).

In the Casual-to-clear speech conversion, if we observe closely, we can see the time-stretching in the casual speech spectrum (top) and a crisp high raising bands in the clear speech spectrum (middle). This difference is because of the vowel space expansion in the case of casual speech and a clearly spoken sentence in the clear speech. The Output spectrum(bottom) closely matches with the target spectrum (middle).

5 Conclusion

In this study, we were able to establish a workflow for end-to-end speech conversion using Bidirectional LSTM based RNN architecture. Moreover, we were able to conclude that the use of LSTM and bi-directionality are a contributing factor when a RNN based approach is used for speech conversion. From our observations we found that the convergence of Casual to Clear speech conversion was twice as fast as that of the male to female voice conversion. This could be due to the sequence lengths in the Casual to Clear speech corpus were shorter.

An important question that still needs to be answered is that we were able to successfully convert one speech into another only after adding the first energy component from a target MGC instead of the source MGC itself. This could be a hindrance when we want to predict a sequence from the model learnt without knowing a priori what the target output looks like.

One of the observations we made, were the Cepstral features required to synthesize the converted speech were very sensitive to any kind of distortions or noise.

6 Future work

This study is a first step towards conversion of Casual speech to Clear speech. In the future, this Recurrent Neural Network model should be able to convert source speech without knowing the actual targets. To that end, one possibility is being able to specify by what percentage we want to expand the casual speech. A generative Bidirectional LSTM based RNN can be useful in that case since we would like to predict sequences without prior knowledge of what the clear speech sounds like. Comprehensive user studies are required for evaluation of the converted speech, both subjective and objective evaluations.

7 REFERENCES

1. B.-H. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.)," IEEE Transactions on Information Theory 32(2), pp. 307–309, 1986.

2. A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, 1, pp. 285–288, IEEE, 1998.

3. T.Toda,A.W.Black,andK.Tokuda,"Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," IEEE Transactions on Audio, Speech, and Language Processing 15(8), pp. 2222–2235, 2007.

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

5. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine 29(6), pp. 82–97, 2012.

6. A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," IEEE Trans- actions on Audio, Speech, and Language Processing 20(1), pp. 14–22, 2012.

7. L.H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 22(12), pp. 1859–1872, 2014.

8. T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets.," in Interspeech, pp. 369–372, 2013.

9. R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," Neural computation 1(2), pp. 270–280, 1989.

10. Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks 5(2), pp. 157–166, 1994.

11. R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks.," ICML (3) 28, pp. 1310–1318, 2013.

12. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation 9(8), pp. 1735–1780, 1997.

13. A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional lstm and other neural network architectures,” Neural Networks 18(5), pp. 602–610, 2005.

14. P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri, “Exploiting the past and the future in protein secondary structure prediction,” Bioinformatics 15(11), pp. 937–946, 1999.15.

15. A. Graves, N. Jaitly, and A. Mohamed, “Hybrid speech recognition with deep bidirectional lstm,” in Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, pp. 273–278, IEEE, 2013.

16. Y. Fan, Y. Qian, F.L. Xie, and F. K. Soong, “Tts synthesis with bidirectional lstm based recurrent neural networks.,” in Interspeech, pp. 1964–1968, 2014.

17. M. W˙ollmer, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, “Feature enhancement by bidirectional lstm networks for conversational speech recognition in highly non-stationary noise,” in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 6822–6826, IEEE, 2013.

18. L. Sun, S. Kang, K. Li, and H. Meng, “Voice conversion using deep bidirectional long short-term mem- ory based recurrent neural networks,” in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pp. 4869–4873, IEEE, 2015.

19. A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pp. 6645–6649, IEEE, 2013.

20. C. M. Bishop, Neural networks for pattern recognition, Oxford university press, 1995.

21. M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing 45(11), pp. 2673–2681, 1997.

22. J. Kominek, A. W. Black, and V. Ver, “Cmu arctic databases for speech synthesis,” 2003.

23. L. Calandruccio and R. Smiljanic, “New sentence recognition materials developed using a basic non-native english lexicon,” Journal of Speech, Language, and Hearing Research 55(5), pp. 1342–1355, 2012.

24. J. Driedger and M. Mu¨ller, “Tsm toolbox: Matlab implementations of time-scale modification algorithms.,” in DAFx, pp. 249–256, Citeseer, 2014.

25. W. Verhelst and M. Roelands, “An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech,” in Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on, 2, pp. 554–557, IEEE, 1993.

26. H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, “Tandem-straight: A temporally stable power spectral representation for periodic signals and applications to

interference-free spectrum, f0, and aperiodicity estimation," in Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pp. 3933–3936, IEEE, 2008.

27. K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation.," in ICSLP, 94, pp. 18–22, 1994.

28. 27. S. Imai, T. Kobayashi, K. Tokuda, T. Masuko, K. Koishida, S. Sako, and H. Zen, "Speech signal processing toolkit (sptk), version 3.3," 2009.

29. D. Erro and A. Moreno, "Weighted frequency warping for voice conversion.," in Interspeech, pp. 1965–1968, 2007.

30. M. Berglund, T. Raiko, M. Honkala, L. K¨arkk¨ainen, A. Vetek, and J. T. Karhunen, "Bidirectional recurrent neural networks as generative models," in Advances in Neural Information Processing Systems, pp. 856–864, 2015.

31. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 , 2014.

32. T. T. D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, et al., "Theano: A python framework for fast computation of mathematical expressions," arXiv preprint arXiv:1605.02688 , 2016.

33. Lasagne, "Lasagne," 2015.Lasagne, https://lasagne.readthedocs.io/en/latest/index.html, 2017