Exploration of Audio Feedback for L2 English Prosody Training

Pedro Miguel Sonso Sousa
IST / ULisboa
Lisboa, Portugal
Email: pedro.sonso.sousa@tecnico.ulisboa.pt

Abstract—**This work explores two different approaches to tackle prosody training in the context of Computer Assisted Language Learning. This would be applied to an exercise where the user listens to a recording containing an utterance made by a native speaker and repeats that utterance, which is evaluated in terms of pitch and duration. The task will be complementing or replacing a reference utterance pre-recorded by native speaker with an utterance in the voice of the learner. The first approach consists on manipulating the user's speech, by correcting the pitch and duration markers through speech analysis. The second approach uses a Voice Conversion method to convert the native speaker's utterances to the voice of the leaner. Both approaches are implemented and preliminary results and evaluations are provided.**

1. Introduction

English can be seen as the lingua franca of the world - it is a global language that strongly dominates all areas of international communication. The number of English speakers may be as high as 2300 million [1], from which the majority of its speakers are non-native. Due to the necessity to accommodate the raising number of learners, learning English has become immensely diversified. The introduction of Computer Assisted Language Learning (CALL) systems provides a cheaper, simpler and more versatile way to learn, because they can be used anywhere at any time and the experience may be tailored to an individual user.

In order to master a language, it is important to master both the individual phonetic segments (vowels and consonants) as well as the properties of syllables and larger units of speech, the supra-segmental aspects of speech. These are known as prosody, and mastering it should part of the focus of CALL systems should be prosody training.

Prosody training through CALL systems can be done both with visual and audio feedback. But since repetition is a key factor on language learning, listening to an utterance from a native professional speaker, would provide the student with a reference to follow. And if the student would listen to himself/herself uttering that same sentence with the correct stress, it would potentially eliminate any distracting factor related with the difference between the student and the native speaker's voices, and improve the student's focus on the real aspects that he/she needs to improve.

The main objective of this thesis is to explore two methods that could provide the student with a reference audio with a voice close to the student's own voice. It is assumed that the student already masters the segmental aspects of the English language. The first approach will consist of manipulating the audio of the user's attempt on this exercise, correcting the pitch and duration of the phones uttered, and playing it back to the user. This will be made using a Vocoder-based system and it is intended to be used without any pre-training and to be available to the user since the first attempt of the first exercise.

The second approach is intended to replace the native-speaker's utterance by the same sentence with the target prosodic contours, but in the user's own voice. It requires gathering data (audio files) from previous exercises from the user to fine-tune a pre-trained Voice Conversion model. This model is then used to generate the reference when the user loads the exercise, in his/her own voice, without requiring any attempt on that exercise. It makes use of state of the art Voice Conversion technology and requires relatively high computational power and long training times.

ELSA Speak. This work was developed in cooperation with ELSA Speak, which is an English learning APP. Its exercises consist of asking the users to read a word or sentence, which is recorded by the device and sent to its servers to be analysed according to targets are defined according to the Western American English accent. It returns real-time feedback on their pronunciation mistakes, with over 95% accuracy. The user may also chose to listen to this word or sentence uttered by a native English speaker.

2. Related Work

2.1. Speech Synthesis

Speech synthesis is defined as the artificial production of human speech. In order to do this, first it is necessary to analyse and decompose the speech signal into parts. These parts can then be manipulated and later synthesized back into speech that is different from the original. This is the backbone of the modern Text-to-Speech (TTS) systems.

2.1.1. Parametric Speech Synthesis. These models focus only on how human-like the output sounds, without making deeper claims that the model is a true model of the human speech production. Human speech is produced as the result of individual parts of the human body, such as the tongue, the lips, the vocal folds and the shape of the vocal tract, combined together, which implies that speech is composed by a number of processes running concurrently. To model this, the components need to be separated to some degree and controlled separately. The current models of these parametric vocoders encode speech into the three following components, according to a given time frame:

- **Spectral Envelope** - Contains the formants originated by the shape of the vocal tract . Perceived as the overall timbre.
- **Fundamental Frequency** - Rate of vibration of the vocal foals. Perceived as the pitch.
- **Non periodic Energy** - Associated with the fricatives.

Usually, these components are expressed in numbers, and vectors of numbers, taken at fixed intervals of time. Once it is in this form, the data can be manipulated for purposes such as Voice Transformation (VT), or used together with the text they represent to train TTS systems. These representations can then be used by a waveform generator to output artificially generated speech.

2.1.2. Neural Vocoders. Since the introduction of WaveNET [2], neural vocoders have gradually became the most common vocoding method to generate waveform audio, achieving increased audio quality of generated speech. These systems are data driven and they do not assume any mathematical model, which appears to be a solution to some inherent problems of the parametric vocoders. One downside is that these models require big amounts of data and take very long to be trained. This makes it difficult to create a universal system that is able to generate any voice even though some attempts already achieve good results, like WaveGlow [3] and Universal Vocoder [4] Also, fine-tuning the algorithm every time the speaker changes is very time-consuming and the inferences usually take long to run, making them not suitable for real-time applications.

2.2. Voice Conversion overview

Voice Conversion (VC) is the study that deals with the conversion of the perceived speaker identity, while retaining the linguistic content.

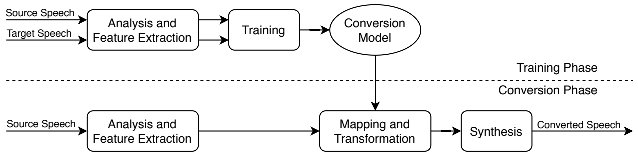

Figure 1. Overview of the flow of a typical Voice Conversion system, divided intro the training and conversion phases

There is a wide range of algorithms designed to perform a VC task, which have different requirements and different purposes. But there are a set of steps that are common to the majority of these algorithms and produce a simplified pipeline that can be seen on the figure 1.

2.2.1. Analysis and Feature Extraction. - Estimation of the parameters that represent the acoustic features of speech, as mentioned in the previous chapter. This is applied to the audio files containing both the source's and the target's speech.

2.2.2. Training. - Receives the features extracted from both the source and target speakers and attempts to represent the relation between similar features of both. Outputs a conversion model, or a mapping function, with the perceived correspondences between each set of features.

2.2.3. Mapping and Transformation. - Performs a process similar to the one on the training phase to map the features of the source speaker into the representations from the conversion model, and performs the transformation of these features using the mapping function. Outputs the converted features.

2.2.4. Synthesis. - Receives the converted features and attempts to reconstruct the waveform audio containing the text originally uttered by the source speaker but with the voice of the target speaker.

3. Pitch Transplant

3.1. Overview

The first approach that was adopted to provide the student with an utterance in his/her own voice is named Pitch Transplant, and the name comes from the way the output is generated. In broad terms, it is built by fusing the user specific features with the scaled and aligned pitch contour of the audio from the reference. It is aligned through a Dynamic Time Warping (DTW) algorithm with specific restrictions so that the produced utterance is close to the target, but doesn't contain significant distortion. This allows the introduction of the gradual reference [5] that will change at each iteration, as alternative to a voice conversion algorithm which will produce a single static target, no matter how the user utters the sentence.

It receives two audio files as input, one from the user and another from the reference, and outputs a single audio file containing the re-synthesized audio of the user with the corrections on pitch and duration contours. All audio files are in wav format, single-channel, sampled at 16KHz and with 16bits per sample

ELSA's server will return the alignment at the phone level of both the reference utterance and the user's utterance. This exercise is expected to only be available to users that already have a good level of English, in terms of phonetics,

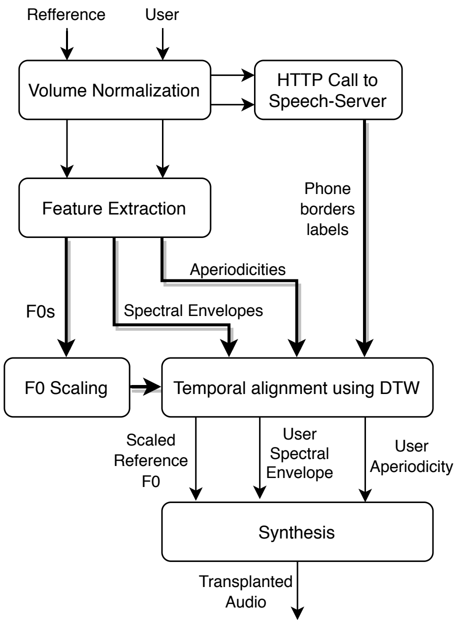

Figure 2. Outline of the Pitch Transplant Algorithm. The thicker lines represent data from both the reference's and the user's utterance.

and consists of uttering a sentence that is shown in the screen.

The outline of the Pitch Transplant algorithm is presented on image 2. The code of the Pitch Transplant algorithm was all written in Python. The original implementation of the WORLD algorithm is written in C++, so a Python Wrapper was used [6].

3.2. WORLD

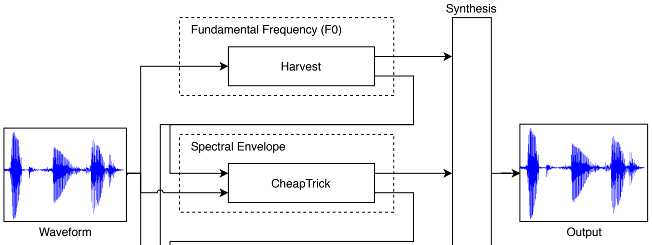

Figure 3. Overview of WORLD, including the methods for estimation of the 3 speech components, Fundamental Frequency, Spectral Envelope and Aperiodicity. Adapted from [7].

WORLD [7] [8] is a vocoder-based high-quality speech synthesis system that allows the easy manipulation of speech and meets the requirements of high sound quality and real-time processing. The version used in this work is the latest, which was used as part of the baseline model of 2020's Voice Conversion Challenge [9].

WORLD can be divided into 2 main stages: feature extraction and synthesis. Feature extraction is carried out by three modules that run in sequence and extract three speech parameters. These are:

- Harvest [10] extracts the Fundamental Frequency (F0).
- Cheaptrick [11] extracts the spectral envelope.
- D4C extracts the aperiodicity [8].

The synthesis stage performs the inverse process, joining the 3 speech components to produce a waveform signal containing speech.

3.3. DTW

To preform the temporal alignment, the DTW algorithm was chosen. This algorithm is fast and computationally cheap, so it fits the requirements for the Pitch Transplant. This implementation of the DTW uses Euclidean Distance. Then, the optimum alignment will correspond to the path in this cost matrix that minimizes the cumulative distances and obeys to a set of restrictions. These restrictions minimize distortion, reduce computational power and ensure the usability of the results. These are:

Endpoint Constaints. specify that the alignment must start in the first frame pair and finish in the last.

Monotonicity Conditions. do not allow for the warping path to have a negative slope.

Global Path Constraints. restrict the region in the matrix where the the distances are calculated and consequentially, the optimal path is searched. This implementation uses a simple Sakoe-Chiba band with the width of either 15 or 31 samples, which is chosen according to the difference between the length of the segments for each phone. In case this difference is smaller than 10 samples, the smallest window is used and the biggest window is chosen otherwise. For the rare cases where there is a difference between segments bigger than 31 samples, the window size is changed to 1.1 times the difference between segments.

Local Path Constraints. specify the allowed jumps between each 2 adjacent elements on the path. It is recommended [12] that the selection of the local continuity constraints should be based on heuristics and observations that result from an experimental process. The allowed steps are represented on figure 4.

- **0**: $D(i, j)$
- **1**: $D(i + 1, j)$ (max of 3 successive steps)
- **2**: $D(i - 1, j)$ (max of 1 successive step)
- **3**: $D(i, j + 1)$ (max of 3 successive steps)
- **4**: $D(i, j - 1)$ (max of 1 successive step)

The DTW returns a variable $wrap\_path$. This variable contains 2 arrays with the indexes corresponding to one of the utterances, reference or user. The correspondence

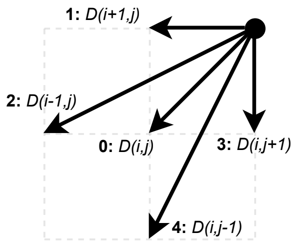

Figure 4. Local continuity constraints. The five step patterns selected.

between the i^{th} frame from the speech patterns X and Y can be obtained by: $X(wrap\_path_X(i)) \Leftrightarrow Y(wrap\_path_Y(i))$

Figure 5 shows the warping path that resulted from the application of Pitch Transplant, and it also contains markings for the beginning and end of each phone. The resulting F0 contours from the utterance synthesized after these alignments are presented on figure 6. The F0 contour from the user's utterance is presented on the top, the one from the reference's utterance on the bottom and in the middle is the F0 contour from the synthesized transplanted utterance.

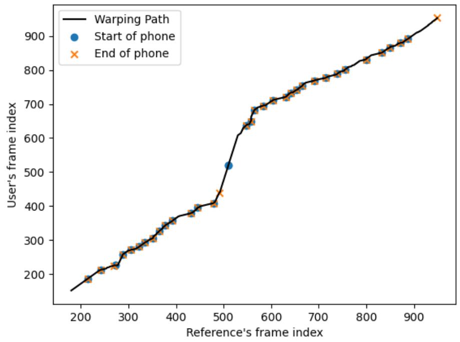

Figure 5. Warping path of an utterance from dataset DS3-5

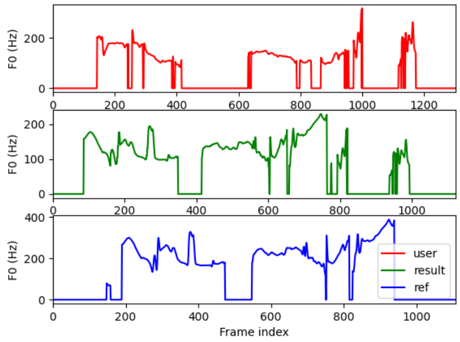

Figure 6. F0 contour of an utterance from dataset DS3-5, the reference utterance and the converted utterance (result of Pitch Transplant algorithm)

3.4. F0 Scaling

The utterances aligned with the DTW algorithm belong to different speakers, so the F0 needs from the reference needs to be scaled in order to be forced into the user's utterance.

The formula used to scale the F0 was adapted from [13]:

$$f0_{final} = \frac{\overline{f0_{user}}}{\overline{f0_{ref}}}(f0_{aligned} - \overline{f0_{ref}}) + \overline{f0_{user}}$$

Where:

σ_{user_s} - Scaled standard deviation of user's F0

σ_{ref} - Standard deviation of reference's F0

$f0_{aligned}$ - F0 contour from the reference aligned with DTW. It is similar to the reference's F0 but with some dropped or repeated frames.

$\overline{f0_{ref}}$ - Mean value of the reference's F0

$\overline{f0_{user}}$ - Mean value of the user's F0

3.5. Signal-to-Noise Ratio (SNR) verification and Feature Normalization

In order to guarantee that the output of the Pitch Transplant does not contain a high level of noise, WADA SNR [14] was introduced to estimate the SNR value. Only utterances with SNR higher than 50 will be processed. It has a python implementation [15] which made it simple to test and integrate into the Pitch Transplant algorithm.

The DTW algorithm receives as input a stack made of both the spectral envelope and the aperiodicity features. But while the aperiodicity values range between 0.001 and 1, the range of the values of the spectral envelope differs greatly in several orders of magnitude, from 10 to 10^{-18}. In order to stack these features, a min-max Normalization was performed, separately on each of the features. It is important to note that this normalization is used only to perform the alignment. The original non-normalized features are kept untouched to perform the synthesis once the time alignment was determined.

3.6. Datasets

In order to test the performance of the algorithm, 3 datasets were created using audios from ELSA's users. All datasets are balanced in terms of gender (50% female and 50% male speakers), with an age that ranges between 18 and 50. The selected utterances have a good level of english pronunciation, as evaluated by ELSA's nativeness score. The characteristics of each dataset are presented on table 1.

3.7. Evaluation

The evaluation methods proposed above take into account mainly the quality of the audio produced by the algorithm and the improvement on the fluency, nativeness and naturalness of the speech.

Pitch Transplant is performed exclusively using CPU and all tests were done in a standard laptop (Intel Core i5-8250U CPU MAX 3.4 GHz, and 8 GB RAM).

TABLE 1. CHARACTERISTICS OF THE DATASETS

Dataset	Nr. of files	Duration of audio (s)
DS1	30	165.2
DS2	20	120.56
DS3-1	10	97.31
DS3-2	10	73.06
DS3-3	10	80.86
DS3-4	10	112.91
DS3-5	10	73.33

3.7.1. AB test. The pair of samples on this test are the original and the transplanted audio. Subjects were asked to listen to both samples and chose which sounds more native, fluent and natural. The neutral option was also given and the sample order was mixed. The results may be found below.

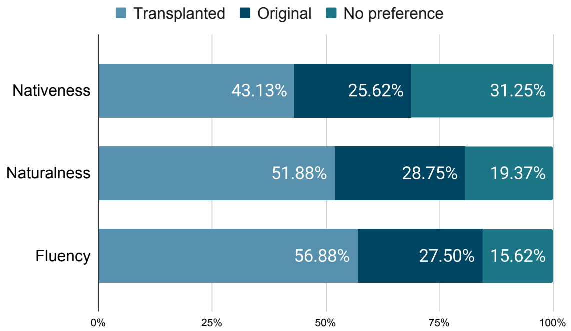

Figure 7. AB tests results

3.7.2. ABX test. The same pair of samples from the AB test were used and a third audio sample (X) containing the reference audio from ELSA's speech artist was added. The subjects were asked to choose from the first pair of samples (A and B) which one was more similar to the reference audio file in terms of intonation, tone, rhythm, and stress. The results may be found in figure 8.

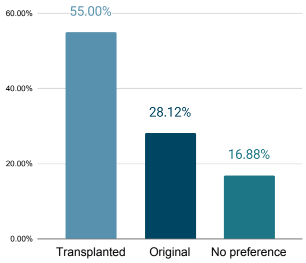

Figure 8. ABX tests results

3.7.3. Mean Opinion Score (MOS). The subjects were asked to rate the audio quality of each of the three samples from 1 to 5. On the user's side, all the samples were either copy synthesized with WORLD, or the result of the Pitch Transplant algorithm. To keep the test centered on the quality of the algorithm, no original samples, without being passed through WORLD, were included. On the reference side, half of the samples were original and half were copy synthesized with WORLD. The objective is to verify how much WORLD reduces the quality of the samples. The results are given in the table below, together with the 95% confidence interval.

TABLE 2. MEAN OPINION SCORE TOGETHER WITH THE 95% CONFIDENCE INTERVAL

Audio Sample	MOS
User Transplanted Audio	2.83 ± 0.14
User Audio synthesized with WORLD	2.89 ± 0.14
Reference Audio synthesized with WORLD	4.39 ± 0.16
Reference Audio (original)	4.72 ± 0.14

3.8. Objective Testing

When performing a Prominence exercise on ELSA app, a prominence marker is calculated for each word in the sentence. These markers result from the evaluation of both the duration and the pitch of each word and return either ”normal” or ”error” whether the submitted recording is close enough to the reference or not. The proposed objective evaluation method makes use of these markers to compute a percentage of how many markers were correct before and after the pitch transplant takes place, calculated with the following formula:

$$avg\_marker\_score(\%) = \frac{\sum correct\_markers}{\sum total\_markers} \times 100$$

This is made for all the utterances of each dataset, resulting in the average marker_score per dataset from DS3 that can be seen in 3.

TABLE 3. RESULTS OF MARKER SCORE TEST (PT - PITCH TRANSPLANT)

Dataset	Before PT (%)	After PT (%)
DS3-1	79.23 ± 5.59	95.0 ± 2.35
DS3-2	87.01 ± 5.15	91.67 ± 6.00
DS3-3	86.12 ± 6.72	93.75 ± 3.65
DS3-4	88.89 ± 4.50	94.0 ± 2.19
DS3-5	88.63 ± 3.96	97.08 ± 2.01
Average	85.98	94.30

3.9. Yes/No question

At the end of the survey, the subjects were asked if they would be comfortable listening to their own manipulated (corrected) voice as a reference in a language learning context. It was a yes or no question, but the indifference option was also given.

4. Voice Converion Approach

This chapter explores an alternative approach to tackle the problem of this thesis. Instead of manipulating a recording from the user, the idea is to produce the user's goal

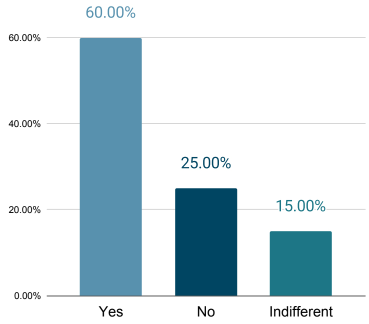

Figure 9. Responses to the question ”Would you be comfortable if, in a language learning context, you would listen to your own manipulated (corrected) voice as a reference?”

utterance in his/her own voice, by way of Voice Conversion (VC), but keeping the prosodic patterns from the source. In this context, the terms reference and user will be replaced by source and target, respectively.

The main objective of this chapter is to generate discussion and to explore the potential of Voice Conversion applied to language learning. No extensive research on all the existing VC alternative methods was made and the chosen algorithm was used with minimum tweaking. The priority was the preparation of a dataset with high prosodic variance, that could be used for the pre-train of a VC model as well as verifying if the user's data from the ELSA app was usable for this purpose.

4.1. Algorithm

4.1.1. Requirements and restrictions. The chosen algorithm to perform this task would need to fit into the requirements below:

itemsep=0pt, parsep=0pt

- **Small amount of target data** - The option of listening to the reference utterance in his/her own voice should be available to the user early on, so the algorithm should require a small amount of the user's audio files.
- **Generate audio files fast** - Even though the training of the algorithm can be made offline, the generation of the audio files should be done fast. Ideally, it should happen during the loading of one exercise, or when an entire module is downloaded. Generating all the converted audio files beforehand and storing them could also be an option, but it is not the current goal.
- **Keeping prosodic features** - The algorithm should allow for the maintenance of the prosodic patterns of the source speaker.

The choice landed on a Non-Parallel Sequence-to-Sequence Voice Conversion Algorithm with Disentangled Linguistic and Speaker Representations, presented in [16]. The algorithm seemed to fit into the first two of the above constraints. The inference process took under 5 seconds per utterance (excluding the waveform generation) and the fine-tune process could converge and achieve decent results with under 200 audio files of the target speaker. As for the third constraint, related with the prosodic features, it will be determined by testing. The code for the Non-parallel Seq2seq Voice Conversion algorithm is provided in the authors github repository.

The algorithm is shipped with an implementation of a Griffin-Lim Vocoder [17] as the waveform generator. To obtain improved results, a pre-trained model of the Universal Vocoder [4] will be used instead.

4.1.2. Preprocessing. The author provided a feature extraction script, extract_features.py, that extracts the mel-spectrograms from the audio files. A peak amplitude normalization and a digital filter to add pre-emphasis to the utterances were added to the feature extraction script. It was added so that the Universal Vocoder pre-trained model could be used, which preforms this pre-emphasis to the input files used on training.

One script, preprocess.py, was created to walk through the dataset directory and generate the files containing the train, validation and test lists. To prevent out of memory errors, all the utterances longer than 7.5 seconds were not added to these lists, as per recommendation of the author.

4.1.3. Architecture. The system performs sequence-to-sequence (seq2seq) voice conversion using non-parallel training data. The process can be broadly divided into two distinct phases, training and conversion. Since the model was used with minimal tweaking, with a black-box approach, the description of the algorithm will not be done in detail. The training phase is responsible for the estimation of the model's parameters and it is done in two stages, the pre-training stage, which uses a multi-speaker dataset, and the fine-tuning stage performed on a specific pair of speakers. The conversion phase receives the acoustic features of the source audio file and converts them to the target using the parameters estimated in the training phase. The converted audio file can then be synthesized by a waveform generator.

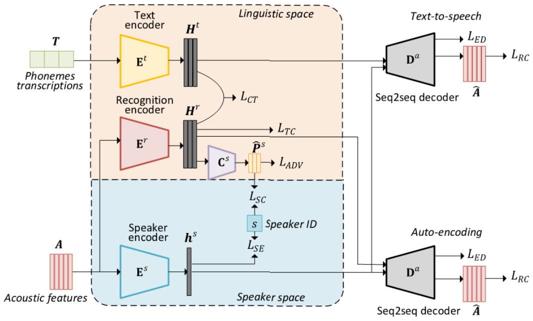

Figure 10. Structure of the Non-parallel Seq2seq Voice Conversion algorithm. Taken from [16]

The model is composed by five main components.

Text Encoder E^t**.** Transforms the text inputs T into linguistic embeddings H^t.

Recognition Encoder E^τ**.** Receives the acoustic feature sequence A and predicts the phoneme sequence T, aligning the acoustic and phoneme sequences automatically. Since one phoneme usually corresponds to tens of acoustic frames, the encoding is a compression process. Its output H^τ has the same length as the phoneme sequence T regardless of the speaking rate of speakers, and resides in the same linguistic space as H^t, containing only linguistic information.

Speaker Encoder E^S**.** Embeds the acoustic feature sequence A into a speaker embedding vector h^S which can discriminate speaker identities and should contain only speaker-related information. It is only employed at pre-training, whereas at fine-tuning stage a trainable speaker embedding is introduced for each speaker, initialized by h^S.

Auxiliary classifier C^S**.** Employed to predict the speaker identity from the linguistic representation H^τ of the audio input. It is introduced for adversarial training in order to eliminate remaining speaker information within linguistic representation H^τ. Each element of its output $\hat{P^S}$ is the predicted probability distribution among speakers.

Seq2seq decoder D^a**.** Recovers acoustic features from the combination of speaker embeddings h^S and linguistic embeddings, H^τ or H^t, either of each fed into the decoder at each training step. It can be viewed as a decompressing process, in which the linguistic contents are transformed back in acoustic features $\hat{A}$, conditioned on the speaker identity information.

A learning rate decay was added to the pre-train script, which was already included in the fine-tune script. In the initial tests, the model kept collapsing, and the results were only noise and silence, so this portion of code was copied from the fine-tune scripts. After this change, the algorithm managed to train successfully.

4.1.4. Waveform Generation. The Universal Vocoder [4] is a WaveRNN-based neural vocoder developed to overcome the over-fitting that other neural vocoders are prone to. It can be used with speakers unseen in training, making it ideal to test in a black-box approach. The authors provide a pre-trained model, making audio generation an easy and seamless process. The pre-train is made with audio files sampled at 16KHz, which is the same sampling frequency used on the VC algorithm.

4.2. Datasets

In order to pre-train and fine-tune the model, it was needed to collect speech data. The datasets used in this task are presented below. The prosodic value is a very subjective assessment based on the textual content and listening of a small subset of the audio files.

TABLE 4. TABLE CONTAINING THE SUMMARY OF THE MAIN CHARACTERISTICS OF EACH DATASET. .

Dataset	Sampling Freq. & Bit Depth	Length of audio min	Prosodic value
VCTK	48KHz/16bit	1640	Low
ARCTIC-rms	16KHz/16bit	66	Medium
ARCTIC-slt	16KHz/16bit	57	Medium
ELSA-REF$^{(*)}$	16KHz/16bit	183	High
LibriTTS	24KHz/16bit	12372	Medium
ELSA-USR1$^{(*)}$	16KHz/16bit	2	High
ELSA-USR2$^{(*)}$	16KHz/16bit	1	High
ELSA-USR3$^{(*)}$	16KHz/16bit	10	High
L2-ARCTIC-NCC	44.1KHz/16bit	70	Medium
L2-ARCTIC-HQTV	44.1KHz/16bit	69	Medium

$(*)$ Non public datasets constructed or adapted for the purpose of this work using protected data from ELSA Corp.

Out of the datasets on table 4, VCTK, ARCTIC-rms, ARCTIC-slt, LibriTTS, L2-ARCTIC-NCC and L2-ARCTIC-HQTV are publicly availible datasets, recorded quiet environments. The remaining datasets were built with audio files from ELSA's speech artist and ELSA's users. Three different detasets were generated with users' audios in order to test different conditions for training, namely the size of the dataset, the fluency of the speakers and the audio quality of the recordings.

ELSA-REF. This dataset was named ELSA-REF because it is composed exclusively by the audio references from ELSA's exercises, recorded by a female speaker. The quality of the audio files is still high and close to studio quality. The sentences of 3 words or less were removed because the existence of many short sentences in the dataset could introduce bias in the Voice Conversion algorithm when used in training.

ELSA-USR1. 13 Audio files from ELSA's assessment test, with a score of 37% of nativeness. The speaker is male and his L1 is Vietnamese. The audios have noticeable background noise and the pronunciation is poor. The audio files needed to be broken into smaller files, due to restrictions of the algorithm, resulting in 23 audio files with a total of 130 seconds of speech.

ELSA-SR2. 13 Audio files from ELSA's assessment test, with a score of 97% of nativeness. The speaker is female, her L1 is American English. The audio files have very low noise and the pronunciation is excellent. The total duration of this subset is 67 seconds.

ELSA-USR3. 170 Audio files from ELSA's exercises. The speaker is male, his L1 is Vietnamese and the pronunciation is poor. The audio files have very different recording conditions. There are three factors that are noticeable in this user's audio files, which can be heard across the majority of ELSA's users: the noise levels vary from barely noticeable to very high, including some audio files where the wind noise is higher than the user's own voice; some audio files contain highly distorted voice, most likely from speaking too close to the headphones microphone; some of the audio files seem to be spoken by a different user (also male). These factors and the problems they lead to will be commented on section 4.3, together with the results from this training.

4.3. Tests

All the tests were performed in a remote AWS EC2 instance, made available by ELSA Corp. This instance had a NVidia Tesla K80 GPU, with 11Gb of memory and CUDA version 11.0.

4.3.1. Pre-Training and fine-tuning with VCTK and ELSA-REF. The first test followed all the recommendations of the authors of the VC model, both in terms of datasets and parameters. The objective was to verify the that the code was stable and the model was training properly. The model was pre-trained with VCTK dataset and fine-tuned with speaker p360 from VCTK and ELSA-REF dataset. The results of the Voice Conversion task were as expected. However, almsot none of the prosodic features from the source speaker were kept. The resulting audio file had all the prosodic traits from the target speaker, thus it was not usable for a context of prosody training.

The VCTK dataset contains utterances mainly from a newspaper, which are predominantly declarative and plain. One interesting thought is that the lack of prosodic variance of this dataset, with which the pre-training and fine-tuning was done, may be responsible for a bias on the algorithm towards generating equally neutral utterances. To test this, a next test was made with different datasets.

4.3.2. Pre-training with LibriTTS. On this test, a new model was pre-trained with a different dataset. The dataset used to pre-train the model on this test is LibriTTS augmented with ELSA-REF files. Similarly to the initial pre-training, all the sentences longer than 7.5 seconds will be removed. Due to the higher number of utterances on the dataset, the pre-training took longer, totaling 105h of clock time and reaching 95k iterations, in the 41st epoch.

4.3.3. Fine-tuning with ELSA-REF and ELSA-USR1. This was the first test done with audio from real users, using one of the datasets created for this purpose. The characteristics of the dataset, containing a very reduced number of utterances, with very poor English pronunciation and low recording quality, make this test an extreme case. The model quickly collapsed and it was not possible to get any useful result from it.

4.3.4. Fine-tuning with ELSA-REF and ELSA-USR2. The model did not collapse, and it trained for 10 hours, reaching over 12k iterations after 50 epochs. The majority of the generated audio files had fluent and intelligible speech, but the perceived speaker identity didn't change, with exception of small portions (words or sometimes only individual phones) that sounded similar to the target speaker. This means that overall, the model failed to perform the voice conversion task, and produced utterances with high pitch fluctuations and with no usability on the context of prosody training.

This test seems to indicate that the dataset of utterances from the assessment test is not enough to generate a meaningful training and validation set. But this time, even though it achieved poor results, the model did not collapse during training and it was possible to generate intelligible utterances.

4.3.5. Fine-tuning with ELSA-REF and ELSA-USR3. The algorithm ran for 37h, reaching 17k iterations after 50 epochs. After generating the audio files, it was clear that the model had trouble converging. The audio files were very long, with over 20 seconds, and had only silence and noise similar to speech with the target's voice, but without uttering any word. This noise seemed like specific phones, mostly vowels, elongated and repeated without any meaningful order. It is now clear that the audio quality has a very high influence on the outcome, higher than expected initially, and may indicate that using the available audio files from real users to train the model will not give any usable result for speakers with these characteristics.

4.3.6. Fine-tuning with ELSA-REF and L2-ARCTIC-NCC. This attempt was done with 100 audio files from L2-ARCTIC-NCC, resulting in 6 minutes of speech. The algorithm reached the 50 epochs at 16400 iterations after running for slightly under 12h, with the alignment graphics indicating that it converged. The converted speech was intelligible but it was not natural. The converted voice had some resemblance with the target speaker, but with an added creakiness that made it sound unnatural. It seems that the model did not have enough data to achieve total convergence and a good result.

This experiment also allowed us to verify the behaviour of the method in the presence of target speakers with poor English speaking skills, which includes not only poor prosody but also mispronunciation errors, namely substitutions, deletions, and additions. The audio files generated through this VC model contained barely any of these mispronunciation errors.

4.3.7. Fine-tuning with ELSA-REF and L2-ARCTIC-HQTV. This fine-tuning was made with ELSA-REF and 150 audios from L2-ARCTIC-HQTV, resulting in 9 minutes of speech. The training ran for 10 hours, reaching 16800 iterations.

The converted voice is very similar to the target and the speech is very natural, even though the fine-tuning was made with under 10 minute of speech from the source speaker. The source's prosody is clearly audible in the converted utterance, but it is not equal, as expected. The variance is much higher than in the previous test done on with VCTK dataset and also higher than the utterances from the target speaker.

The algorithm mostly retains the pronunciation, and even the accent, of the source speaker. There are some phones that hint a slight mispronunciation from the target speaker, more specifically deletions, but this mostly appears to be cases where the phones are too short rather than not present. This is a very interesting application for English learning exercises, such as the ones found on ELSA Speak mobile app. It may allow the speaker to hear himself/herself with an almost native pronunciation, either as a reference for the exercise, or as a motivational feature.

With the results obtained in this test, it is possible to move on to the evaluation of the algorithm.

4.4. Evaluation

A survey was set up including some of the audios generated in the last test preformed with the VC algorithm. The survey was responded by 40 different subjects.

4.4.1. MOS. Three scores are calculated and presented in the table below. These 3 scores correspond to three different questions. The first two questions were made after presenting the subjects with three audio samples. A - Sample from ELSA-REF; B - Sample from L2-ARCTIC-HQTV; C - Converted sample. The third question was made to the subjects after presenting them with 3 different converted utterances.

itemsep=0pt, parsep=0pt

1) Would you say that sample C imitates the duration and intonation pattern of sample A?
2) On a scale of 1 to 5, would you say that sample C retains the voice of sample B?
3) How native do these samples sound when compared to an American English Native Speaker?

TABLE 5. Mean Opinion Score and 95% confidence interval of the responses to 3 evaluations of the converted audio according to 3 different metrics

Metric	MOS
Retention of prosody patterns from source	3.31 ± 0.15
Voice similarity to target speaker	3.46 ± 0.16
Nativeness, comparing to American English Accent	3.45 ± 0.25

4.5. AB test

The subjects were presented with 3 pairs of audio samples. Each pair contained one sample taken from L2-ARCTIC-HQTV dataset and one converted sample. They were instructed to listen to each pair and chose which utterance had better English pronunciation.

4.6. Yes/No questions

Two extra questions were made on this survey. The first question was made in relation to the same audio files that were used in the 3rd MOS. The second question was general and did not involve any audio sample. The questions were

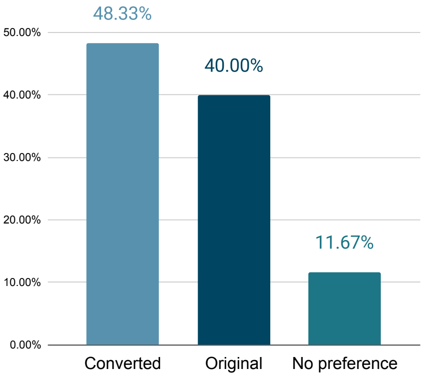

Figure 11. AB tests results for evaluating pronunciation

1) Would you consider that these samples have enough sound quality to be used as a reference in an English language learning exercise?
2) Would you be comfortable if, in a language learning context, you would listen to your voice saying a sentence you never said before?

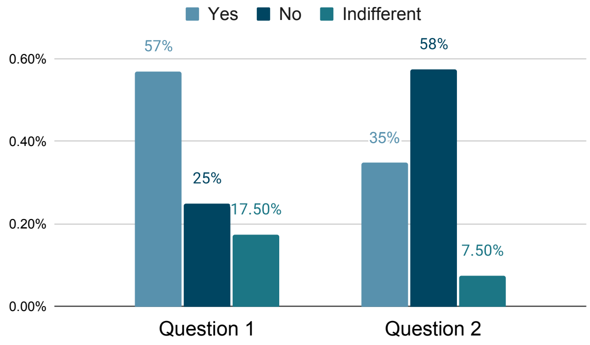

Figure 12. Answers to the yes or no questions presented to the subjects on the online survey

5. Conclusions

The objective of this thesis was to perform an exploratory study of the application of two different techniques to prosody training. It was done in the context of the exercises available in the ELSA Speak app.

The first approach, named Pitch Transplant, presents real-time results since the first use, but it is fairly limited. But since the output is generated by manipulating the input audio, it will maintain its pronunciation mistakes and audio quality. Also, there is a limitation to how much the audio can be manipulated without losing the speech naturalness, allowing only for small tweaks.

The second approach uses a more current technology and still has huge potential to improve. It is a computationally heavy process, requires lengthy pre-training and does not cope well with noisy recording environments. Also, the algorithm requires further development in order to apply it in real-time. Nevertheless, it not only produces more natural speech, but once the model is trained, it may convert virtually any sentence without any further input from the user.

On another note, the surveys that were handed out to evaluate both methods contained extra questions. The answers to these questions revealed that the majority of the subjects were comfortable in having their voices manipulated and used in a language learning context. This encourages further the application of such systems in English learning apps such as ELSA Speak, or even as complementary work for English language courses.

5.1. Future Work

5.1.1. Pitch Transplant. Pitch Transplant manipulates the user's audio, so it is prone to the audio quality of the input audio. Reducing noise would definitely improve results, which today can be done with systems such as Krisp, developed by NVidea.

In order to evaluate the impact of the gradual reference, it is necessary to use it in a language learning context, which would require continuous testing for several months. Due to time limitations this evaluation could not made. But changing the implementation and allowing for remote testing could allow for this test to be performed, which would be an interesting work in the future.

5.1.2. Voice Conversion. In all tests where the model was fine-tuned with audio files from real users, it either collapsed or failed to converge, most likely due to the low quality of the users' audio files. Using an SNR estimator, such as [14], it would be possible to select the audio files with less noise and run the fine-tune with them. Applying a noise reduction system, such as Krisp, could also allow for the application of this model to the user's audios.

During the INTERSPEECH 2020 conference, new VC methods were proposed that could possibly produce better results than the chosen algorithm [18]. One paper [19] focuses its work in developing a technique to transfer the source speaking style in a non parallel voice conversion task . Its performance is better than two baseline models, one of which is the selected model in this work.

It would also be interesting to productize this model and apply it in a real environment where the progress of the students could be monitored. The study would consist of having a group A doing prosody training with the reference sentences in their own converted voices, and group B practicing with the native speaker's audio. After a period of time, the progress of each group of students would be analyzed and compared.

References

[1] D. Crystal, *The Cambridge Encyclopedia of the English Language*, 3rd ed. Cambridge University Press, 2018.

[2] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[3] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," 2018.

[4] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards achieving robust universal neural vocoding," 2019.

[5] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Communication*, vol. 51, no. 10, pp. 920–932, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2008.11.004

[6] C.-C. Hsu, "Pyworld - a python wrapper of world vocoder," https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder.

[7] 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. E99D, no. 7, pp. 1877–1884, Jul. 2016.

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

[9] P. L. Tobing, Y.-C. Wu, and T. Toda, "Baseline system of voice conversion challenge 2020 with cyclic variational autoencoder and parallel wavegan," 2020.

[10] M. Morise, "Harvest: A high-performance fundamental frequency estimator from speech signals," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2017-August, pp. 2321–2325, Jan. 2017.

[11] ——, "Cheaptrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Communication*, vol. 67, pp. 1–7, Jan. 2015.

[12] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. USA: Prentice-Hall, Inc., 1993.

[13] C. Magariños, P. Lopez-Otero, L. Docio-Fernandez, E. Rodriguez-Banga, D. Erro, and C. Garcia-Mateo, "Reversible speaker de-identification using pre-trained transformation functions," *Computer Speech and Language*, 2017.

[14] C. Kim and R. M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008.

[15] J. Meade, "Wada snr estimation of speech signals in python," https://gist.github.com/johnmeade/d8d2c67b87cda95cd253f55c21387e75, 2020.

[16] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, no. 1, pp. 540–552, 2020.

[17] D. Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[18] S. Ding, G. Zhao, and R. Gutierrez-Osuna, "Improving the speaker identity of non-parallel many-to-many voice conversion with adversarial speaker recognition," in *INTERSPEECH*, 2020.

[19] S. Liu, Y. Cao, S. Kang, N. Hu, X. Liu, D. Su, D. Yu, and H. Meng, "Transferring source style in non-parallel voice conversion," 2020.