TÉCNICO
LISBOA

Text-to-Speech Synthesis in European Portuguese using Deep Learning

Ana Catarina Rosa Gonçalves

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Dr. Isabel Maria Martins Trancoso
Dr. Sérgio Manuel Gaspar Ferreira Paulo

Examination Committee

Chairperson: Prof. Dr. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Dr. Isabel Maria Martins Trancoso
Members of the Committee: Dr. Helena Gorete Silva Moniz

November 2018

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First of all, I would like to thank my supervisor, Professor Isabel Trancoso for all the guidance and knowledge transmitted, and also for the help given throughout this journey.

For the technical part, I would like to thank Sérgio Paulo for the incredible support and guidance provided, which has been essential to the development of most of the work developed.

To all my friends who were always present, thank you for the encouragement and support. To OAUL and AOAL for giving me the opportunity to do the thing I like the most and for making most of my Wednesdays for the last 3 years very special.

Finally a special thanks to my family. To my parents, for all of the sacrifices that you have made on my behalf, always being there for me and making it possible for me to achieve my goals, and for all the support you gave me throughout this long and exhausting phase.

Last, but definitely not least, a especial thanks to my boyfriend, who always supported me along this journey. His constant encouragement, specially in the demotivating moments, gave me the strength to pursue and accomplish this work.

Abstract

This thesis has one main goal: to synthesise speech from text for European Portuguese, using recent deep learning techniques. The motivation was to use this work on one hand as a first step for building a much needed child's voice, and on the other hand as a framework to synthesise expressive speech. These are two limitations that are faced by synthesizers in many areas of application, in particular, in the interaction with robots and serious games. This topic involves many areas of study such as machine learning, speech synthesis, linguistics and speech acquisition. This article reports the work done, the framework (Merlin) used to synthesise speech and the final conclusions of this project. The choice of Merlin was guided by the target application, the previous experience of the team, and the available information about each method, within the range of speech synthesis methods involving deep learning. The work done with Merlin will enable the synthesis of a child's voice depending mainly on the recordings of a child.

Keywords

Speech Synthesis; Text-to-speech Systems; Portuguese Language; Deep Learning; Child's Voice; Expressive Speech Synthesis

Resumo

Esta dissertação de mestrado tem como principal objetivo a síntese de fala a partir de texto para o Português Europeu utilizando técnicas de "deep learning". A motivação para este trabalho é em primeiro lugar a de construir uma voz de criança que é bastante necessária e por outro lado para o desenvolvimento de uma ferramenta de síntese de fala expressiva.

Nos sintetizadores há duas limitações sentidas em muitas áreas de aplicação, em particular na interação com robôs e jogos sérios. Este tema inclui diversas áreas de estudo tais como, aprendizagem automática, síntese de fala, linguística e aquisição de fala.

Este relatório descreve o trabalho realizado, a ferramenta (Merlin) utilizada para síntese de fala e as conclusões finais deste projeto. A escolha do Merlin, foi guiada pela aplicação desejada para o projeto, pela experiência prévia da equipa, e pela informação disponível sobre cada um dos métodos, dentro do grupo que envolve "deep learning".

O trabalho realizado com o Merlin irá permitir a síntese de fala com voz de criança, estando maioritariamente dependente de se gravar uma criança.

Palavras Chave

Síntese de Fala; Sistemas de conversão texto-para-fala; Língua Portuguesa; "Deep Learning"; Voz de criança; Síntese de fala Expressiva

Contents

List of Figures

List of Tables

Abbreviations

AAC Augmentative and Alternative Communications

AV Arousal-valence

ANN Artificial neural network

ARSG Attention-based Recurrent Sequence Generator

CBHG 1-D convolution bank + highway network + bidirectional GRU

CAT Categorical

CNN Convolutional neural network

DNN Deep Neural Network

EM Expectation–Maximisation

GRU Gated Recurrent Uni

g2p Grapheme-to-phoneme

HMM Hidden Markov Model

IPO Instituto Português de Oncologia

LSTM Long short-term memory

NN Neural Network

MDS Multidimensional scaling

MLSA Mel log spectrum approximation

MOS Mean Opinion Score

MRHSMM Multiple regression hidden semi-Markov model

RNN Recurrent Neural Network

SAMPA Speech Assessment Methods of Alphabet

seq2seq sequence-to-sequence

TTS Text to Speech

1

Introduction

Contents

1.1 Topic Overview

The state of the art of speech synthesis evolved over time, allowing us to distinguish four main generations of Text to Speech (TTS) systems. The generations are, from the first to the newest, synthesis by rule, by concatenation, statistical parametric speech synthesis and deep learning. The block structure is the same for the first three generations, as it can be seen in Figure 1.1.

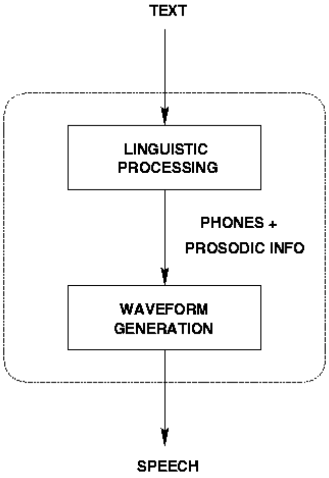


Figure 1.1: Block Structure for TTS systems of the first three generations.

TTS synthesis is based on the generation of a speech waveform to convert standard language text into speech. It is usually made of a front-end and a back-end. The first one has two main purposes, normalisation or pre-processing that converts abbreviations or numbers from the raw text input into words and text-to-phoneme or Grapheme-to-phoneme (g2p) conversion which consists in assigning phonetic transcriptions to each word and dividing and marking text into prosodic units. Examples of prosodic units are phrases, clauses and sentences. The back-end is usually a synthesizer that converts the linguistic features into speech, taking into account the target prosody, such as pitch contour and phoneme duration.

For synthesis by rule, the inputs are phonemes and stress marks, and the output is a continuous waveform. As it can be seen in Figure 1.2, the method consists of a module of synthesis strategy that has stored information about phonemes and rules describing the mutual effects of adjacent phonemes.[2]

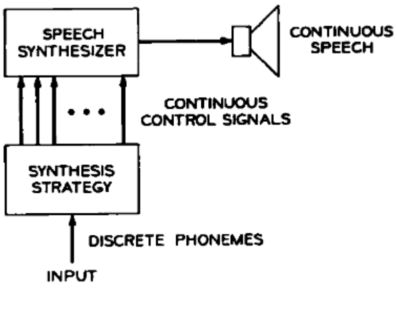


Figure 1.2: Technique of Speech Synthesis by Rule.

Concatenative synthesis is based on recording speech, storing it in a database and then concatenating pieces in order to have the desired output. This method may produce very good results. However, it is very difficult to model expressiveness in speech and the techniques for autonomous segmentation of the waveforms can cause errors in the output. [3]

In order to solve the possible errors in the output caused by concatenation, a new approach was developed, called statistical parametric speech synthesis, that is based in Hidden Markov Model (HMM) models. [4, 5] In this method, the parametric representation such as spectral and excitation is extracted and then modelled with the HMMs.

Figure 1.3 shows a block diagram of an HMM-based speech synthesis system. This system can be divided into training and testing. Training is done by using the Expectation–Maximisation (EM) algorithm that applies the maximum likelihood estimation in order to estimate the model parameters from which the speech waveform is reconstructed. The reconstruction step is made in the synthesis part. This performs a maximisation of the calculated probabilities in the EM algorithm, from a set of estimated models. At last, the speech waveform is synthesized with a speech synthesis filter such as Mel log spectrum approximation (MLSA) and an excitation generator to generate excitation parameters.

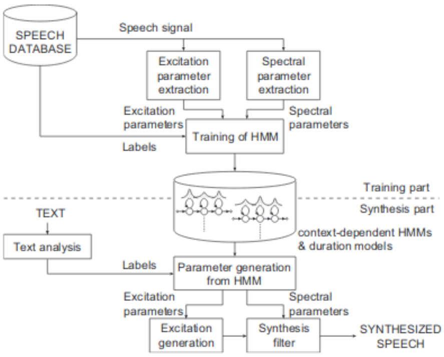


Figure 1.3: Block-diagram of HMM-based speech synthesis system (HTS).

With the progress of computational power and with the recent advances in machine learning, Deep Neural Network (DNN) started to be used to substitute HMMs despite both of them being used nowadays. [6] The main difference between both methods is that while for decision trees in HMM-based systems, the operations are usually at a state-level, and there are separate trees to deal with separate acoustic streams, in DNNs the training is done so the system can predict simultaneously for all the streams at a frame-level.

This signalled the beginning of the 4th generation, as deep learning evolved and the authors started to insert neural networks in the already existent systems. Later they started to develop systems based completely in DNNs until they reached the end-to-end systems.

A DNN is an Artificial neural network (ANN) composed of multiple hidden layers that can model complex non-linear relationships and generate compositional models. This architecture can have many variants. For speech synthesis applications they are usually feedforward networks with Recurrent Neural Network (RNN)s, such as Long short-term memory (LSTM) and also Convolutional neural network (CNN)s.

RNNs can be uni-directional or bidirectional. Its input can have an arbitrary length as well as the output. The main difference between them and the usual DNN or CNN is the contextual information. In other neural networks types, only the past time instances are taken into account. However for the RNNs the neural network can learn from information propagated both forward and backwards in time. This type of DNNs are widely used in synthesis methods.

A CNN is a feed-forward ANN variation of multilayer perceptrons that often use a minimal amount of preprocessing for the input of the net. The input has to have a fixed size, and the CNN uses connectivity pattern between its neurons, generating an output with a fixed size too. This type of DNNs are highly applied in image processing and speech recognition.

One of the goals of this thesis is to develop a 4th generation TTS system for European Portuguese, replacing the 3rd generation in house system known as DIXI. Rather than doing it for a typical adult voice, however, a second goal is to build a synthetic child's voice. A third and overly more ambitious goal is to be able to increase the expressiveness of the synthetic voice.

When talking about speech synthesis, two different characteristics have to be evaluated, intelligibility and naturalness. The first one is the ability of being understood and describes the clarity of the synthesized speech. The second one assesses information that is not included in intelligibility, such as the listen easiness, stylistic consistency and nuances level. [7]

With today's TTS systems, intelligibility is no longer a major issue. However, in order to evaluate naturalness, subjective methods are typically required. Mean Opinion Score (MOS) is commonly used for audio, video or audiovisual quality and is expressed as a number between 1 to 5, 1 being the lowest quality and 5 the highest. See Table 1.1 for the labels used in this subjective test.

Table 1.1: MOS rating scale

Rating	Label
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

The overall MOS score is calculated by the following expression:

$$MOS = \frac{\sum_{n=0}^{N} R_n}{N}. \tag{1.1}$$

where N is the number of subjects that participate in the evaluation and R_n the rating given by each of them. https://siaiap32.univali.br/seer/index.php/acotb/article/viewFile/5314/2776

1.2 Motivation

TTS systems have been in development for a long time with a lot of different applications. Usually the voices used or synthesised are from adults or when a child's voice is synthesised, it is usually very robotic, inexpressive and does not sound genuine because of the difficulties faced when recording a child and when developing a system to synthesise voice that sounds natural. As it will be explained in this report, several methods are now being developed and tested in order to have at least one that accomplishes the purposes of having a natural human voice in a robot device.

There are many applications for TTS systems with a child's voice, the most pressing one as Augmentative and Alternative Communications (AAC) device. The AssistiveWare project [1] is an example of this type of application. Serious games are also an application area for TTS systems with a child's voice. The ECIRCUS[8, 9] project is an example of this type of application. The project had as a goal the development of a serious game to help children aged 9 to 12 with bullying problems.

A third type of application can be found in human-robot interaction. Currently in Portugal there are several systems under development or already in use, that involve the interaction of robots with children. Monarch[2] is a friendly robot used at Instituto Português de Oncologia (IPO) to interact with children that are there for treatment. Another national project is INSIDE[3] that involves the interaction with children with Autism Spectrum Disorder.

The purpose of this thesis is then to first develop a voice in European Portuguese with deep neural network. child's voice that can be applied in the robots used for the Monarch and the INSIDE projects in order to have interaction with the children with a child's voice. This may increase the help given to these children reducing the apprehension felt by them since the communications are done through a voice from an equal.

1.3 Innovations of the work

This research has one innovation for the Portuguese Speech which is the synthesis using neural networks. Usually the speech synthesis in European Portuguese was made from word concatenating methods.

1.4 Thesis Outline

This report is organised in seven chapters. Chapter 1 is the Introduction, where it is given a broad overview of methods used to build learning models for speech synthesis. Then, it is defined which goals are proposed to achieve and the motivation to accomplish them. I also introduced a subsection referring the innovations of this project. It ends with this section, describing this report's structure.

Chapter 2 is entitled Background. There, all the methods studied are presented in order to understand how speech synthesis is currently done. In the end of this chapter we made a brief comparison between each one of the compared methods.

Chapter 3 is about Expressive Synthesis where some projects and methods are explained and Chapter 4, describes all the work done with Merlin. There, it is described the installation, trainings

[1] http://www.assistiveware.com

[2] http://monarch-fp7.eu/

[3] http://www.project-inside.pt/

and the datasets used in English and Portuguese presenting some conclusions of the English speech synthesis.

Chapter 5 is where the results are presented as well as the evaluation form constructed in order to receive the results. The results are described and we present some first conclusions.

In chapter 6 we present a proposed suggestions for future work and finally, chapter 7 is Conclusions, in which some relevant outcomes from this report are presented.

2 Background

Contents

In this chapter, it will be presented methods based on neural networks used nowadays for speech synthesis. As explained before, until now there have been developed for speech synthesis four main generations, synthesis by rule, by concatenation, statistical parametric speech synthesis and deep learning. The work will be focused on the 4th generation, deep learning, because, even if it is the one in the most "embryonic" state, it is already proven that it is the one with the best results and potential. The five methods that will be studied are Merlin, WaveNet, Deep Voice, Char2Wav and Tacotron. Expressive synthesis and the projects related to it will be presented in Chapter 3 as well as the corpus where this thesis will be based in.

2.1 Merlin

Merlin [10] is a neural network-based speech synthesis system based on statistical parametric speech synthesis. It must have a front-end in order to process text and a vocoder that creates the final audio wave files.

The front-ends that are currently used are Festival and Ossian since their outputs are HTS-style labels with state-level alignment. These labels are converted into vectors of binary and continuous features for the input of the neural network.

For vocoder, the system supports currently STRAIGHT [11, 12] and WORLD [13]. STRAIGHT uses F0 extraction [14] while WORLD uses DIO algorithm in order to estimate F0 and CheapTrick [15, 16] for spectral envelop estimation. For synthesis, WORLD uses excitation signal and spectrograms in order to calculate vocal cord vibration using PLATINUM, an aperiodic parameter extraction algorithm.

For training the neural network, the features have to be normalised. Merlin recurs to min-max method or mean-variance. The first method normalises features to the range of [0.01 0.99] and the second normalises features to zero mean and unit variance. Usually with, linguistic features it is used min-max normalisation and for output acoustic features the mean-variance.

The acoustic model has several possible implementations such as feedforward neural networks, Long short-term memory (LSTM) based Recurrent Neural Network (RNN), bidirectional RNN or some variants implemented in Merlin like Gated Recurrent Uni (GRU)s and LSTM.

As it can be seen in Figure 2.1, in a feedforwad neural network in order to predict the output, the input is used by several layers of hidden units, that perform a nonlinear function. The input are linguistic features, and the hidden layers can be activated with a sigmoid and hyperbolic tangent. The output are vocoder parameters.

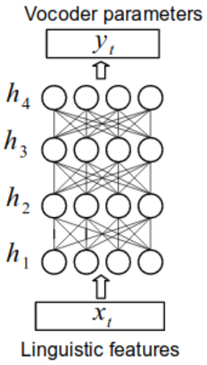


Figure 2.1: *Feedforward neural network with four hidden layers.*

A feedforward neural network differs from an RNN in the way linguistic features are mapped. The first one does not consider the sequential nature of speech and maps the features frame by frame while the second map the features with the sequence-to-sequence (seq2seq) method. Using LSTM units it can be reproduced the behaviour of an RNN. As it can be seen in Figure 2.2 a standard LSTM inputs the signal and the *hidden activation of the previous time instance* through the input gate, forget gate, memory cell and output gate in order to produce the activation.

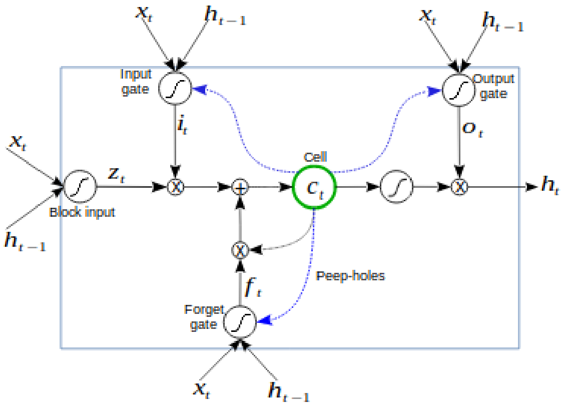


Figure 2.2: Long short-term memory unit. The inputs to the unit are the input signal and the hidden activation of the previous time instance. [10]

As it was explained before in Chapter 1.1, a bidirectional RNNs learn from information propagated forwards and backwards. For Merlin, the hidden units used were bidirectional LSTM-based RNNs.

For training, there were used 2400 utterances 70 as a development set, and 72 as the evaluation set. Festival was the front-end used and for vocoders were STRAIGHT and WORLD.

The audio generated by Merlin is presented online. [1] These samples were obtained using the WORLD vocoder and compared with synthesised samples from STRAIGHT vocoder. The audio presented in the website has better quality than the one produced with the STRAIGHT vocoder in terms of both intelligibility and naturalness as well as speaker similarity.

[1] https://cstr-edinburgh.github.io/merlin/demo.html

2.2 WaveNet

WaveNet is a probabilistic and autoregressive model with distribution for the prediction that generates audio waveforms. It has the ability to know the characteristics of different speakers and switch between them.

WaveNet is based in casual convolutions which guarantee the order of modelling the data. As shown in Figure 2.3 the prediction made at a certain timestep cannot depend of the future. With this the output on a normal convolution is shifted some timesteps and because the timesteps are known the conditional predictions can be made in parallel. For the generation the predictions are sequential so the next sample to predict has into account the predictions already made.

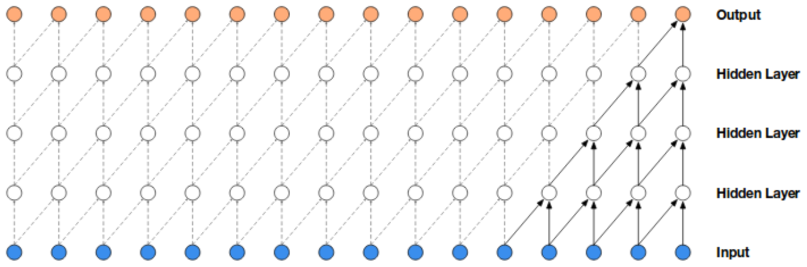


Figure 2.3: Stack of causal convolutional layers.

In order to raise the receptive field without increasing computational cost, it is used dilated convolution in WaveNet. This is a convolution with a filter applied that allows the network to work on a different scale. As it can be seen in Figure 2.4, it is shown stacked dilated causal convolutions. This allows to have a few layers without having vast receptive fields in the network keeping the resolution and computational efficiency.

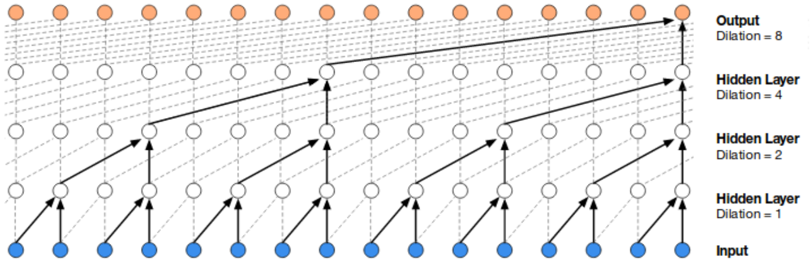


Figure 2.4: Visualization of a stack of dilated causal convolutional layers.

Softmax distribution was used as conditional modelling since it works for all types of data that can be implicitly continuous. This happens because is more flexible and can model arbitrary distributions as it does not make assumptions about its shape.

For the gated activation it was used in the same unit as PixelCNN [17] since it has a non-linear activation function and was concluded that this is better than a rectified linear activation function for audio modelling.

In order to speed up convergence both residual and parameterised connections were used in the network. The residual block is shown in Figure 2.5.

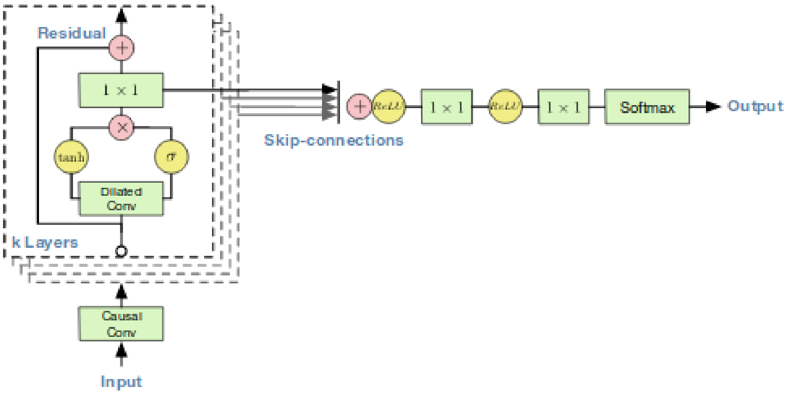


Figure 2.5: WaveNet residual block.

The model is conditioned in the input variables so it can be produced audio with the characteristics in need. For a Text to Speech (TTS), it is required to have as input, information about the text. In WaveNet was used global and local conditioning. The first one influences the output distribution along time steps while the second has a time series, with lower sampling frequency than the audio, that is first transformed with a transposed convolutional network in order to have the same resolution.

For testing WaveNet for the TTS task the authors used Google's North American English and Mandarin Chinese databases which have 24.6 hours and 34.8 hours in the respective dataset. The results were compared with a Hidden Markov Model (HMM)-driven unit selection concatenative and with an LSTM-RNN-based statistical parametric that was used as a base example and model for speech synthesis. Both the example-based models were trained with the databases used for WaveNet so they could be compared.

In order to evaluate the results obtained[2] there were conducted subjective paired comparison tests and Mean Opinion Score (MOS) tests. In the first tests, the subjects were asked to choose from a pair of samples, the one they preferred. The MOS test was conducted as explained in Chapter 1.1 As it can be seen in Table 2.1 the MOS achieved by WaveNet in naturalness above 4.0 which is better than the other systems.

Table 2.1: MOS of speech samples from LSTM-RNN-based statistical parametric, HMM-driven unit selection concatenative, and proposed WaveNet-based speech synthesizers, 8-bit μ-law encoded natural speech, and 16-bit linear pulse-code modulation (PCM) natural speech.

	Subjective 5-scale MOS in naturalness	
Speech samples	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

[2] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

2.3 Deep Voice

Deep voice is an end-to-end TTS system based on Deep Neural Network (DNN). It can be divided in five principle models, as shown in Figure 2.6, segmentation, Grapheme-to-phoneme (g2p) conversion, phoneme duration prediction, fundamental frequency prediction and audio synthesis.

First, the g2p model converts characters from the system inputs to phonemes. For the g2p it is used a multi-layer bidirectional encoder with a GRU and a deep unidirectional GRU decoder.

The segmentation model has the function that locates phoneme boundaries in the voice dataset reaching a phoneme-by-phoneme transcription and identifies where each phoneme begins and ends. This is trained in order to output the alignment between an utterance and a sequence of phonemes. It was adapted a convolutional RNN with the purpose of detecting the boundaries.

The phoneme duration model predicts the temporal duration of every phoneme in an utterance while the fundamental frequency model predicts the voiced phonemes and when it is voiced detects its F0 throughout its duration. This model has as inputs a sequence of phonemes with stress and consists of two connected layers followed by two unidirectional recurrent layers with GRU cells and at last an output layer. This last layer outputs for every input the phoneme duration and the probability of being voiced and a set of time-dependent F0 values.

The audio synthesis model combines the results of the last described models outputs, g2p, phoneme duration, and fundamental frequency prediction and synthesises audio required. This model is based in WaveNet, however, has an encoder with a stack of bidirectional quasi-RNN layers [18] and after the encoding of the inputs is performed its upsampling.

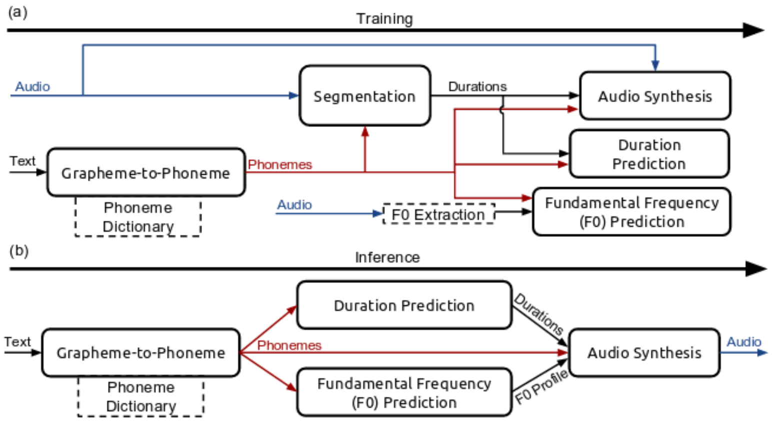
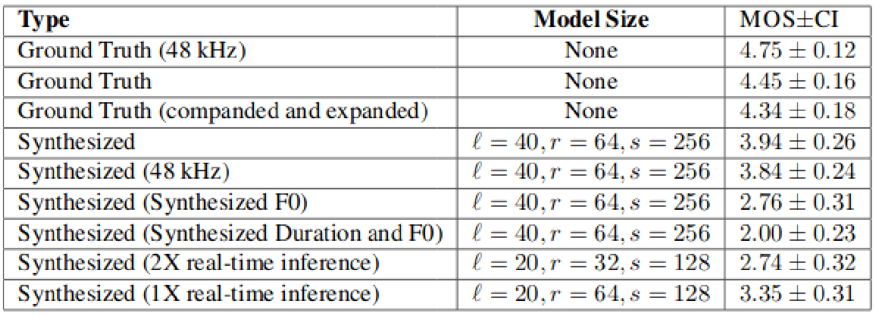


Figure 2.6: Deep Voice system diagram.

Deep Voice was trained with approximately 20 hours of speech data segmented in 13.079 utterances. [19] It was conducted a variety of tests in training the model, using a different number of layers. It was then concluded that below 20 layers the quality of the audio synthesised was poor. Above the 20 layers, the audio was easily recognised and classified with high quality. The higher the number of layers less is the noise in the audio.

In order to conclude the quality of the synthesised audio[3], it was conducted the MOS rating using the CrowdMOS toolkit and methodology [20]. As it can be seen in table 2.2 are presented the scores for synthesis results of Deep Voice and the variations performed.

Table 2.2: MOS and 95% confidence intervals (CIs) for utterances.



Type	**Model Size**	MOS$\pm$CI
Ground Truth (48 kHz)	None	4.75 ± 0.12
Ground Truth	None	4.45 ± 0.16
Ground Truth (companded and expanded)	None	4.34 ± 0.18
Synthesized	$\ell = 40, r = 64, s = 256$	3.94 ± 0.26
Synthesized (48 kHz)	$\ell = 40, r = 64, s = 256$	3.84 ± 0.24
Synthesized (Synthesized F0)	$\ell = 40, r = 64, s = 256$	2.76 ± 0.31
Synthesized (Synthesized Duration and F0)	$\ell = 40, r = 64, s = 256$	2.00 ± 0.23
Synthesized (2X real-time inference)	$\ell = 20, r = 32, s = 128$	2.74 ± 0.32
Synthesized (1X real-time inference)	$\ell = 20, r = 64, s = 128$	3.35 ± 0.31

[3] http://research.baidu.com/deep-voice-production-quality-text-speech-system-constructed-entirely-deep-neural-networks/

2.4 Char2Wav

Char2Wav [7] is an end-to-end model for speech synthesis developed in *Université de Montréal*. It is made up by two principle models, the reader and a neural vocoder. The reader is made by an encoder that is basically a bidirectional RNN that receives as input, text or phonemes. This creates as output the phoneme sequence to be generated. The output of the encoder is received by a decoder with attention, that is the second part of the reader as shown in Figure 2.7. This decoder consists of an RNN with attention. This is an Attention-based Recurrent Sequence Generator (ARSG) that is based in an RNN which receives the phonemes generated by the encoder and creates as output a sequence of acoustic features. For this project, it was used a location-based attention mechanism developed by Graves (2013) [21].

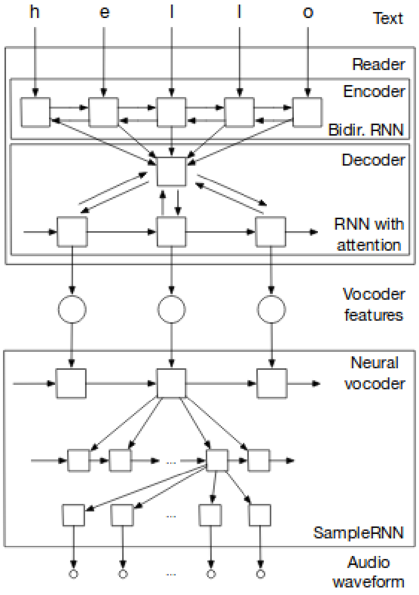


Figure 2.7: Char2Wav: An end-to-end speech synthesis model. [7]

For the second part of this model, it can also be seen in Figure 2.7, speech is synthesised with a parametric neural module based in SampleRNN [22] (Mehri et al., 2016). SampleRNN is a method that uses RNNs so in the training on short sequences, it models longer-term dependencies in audio waveforms [7, 22]. It has a hierarchical structure divided between models, autoregressive multilayer perceptrons, and stateful RNN working in different temporal resolution. With that, it becomes able to learn modifications in temporal sequences, even in the ones with a long duration. This turns the system more efficient in regarding during training.

In Char2Wav, was used a conditional version of SampleRNN in order to relate the sequence of vocoder features with the corresponding audio. It is defined as an extra input each frame of these features in each state of the top layer. This way not only the past samples are used to generate the current audio but also frames of the vocoder features.

In this model, the reader and the neural vocoder were trained separately in a first phase using normalised vocoder features for WORLD vocoder as targets for the first block and inputs for the second one. At last, adjustments were made in order to have the whole model.

Until now the samples available are in Spanish, and the model is being trained with English and German. In order to test Char2Wav, the existing samples are selected from ten random sentences never seen by the model, so some of them fail. According to the investigators, it is difficult to train this model, and this can be caused by a failure in the attention part of the decoder. [4]

2.5 Tacotron

Tacotron [1] is based in a seq2seq model with attention. As it can be seen in Figure 2.8, the model consists in an encoder, an attention-based decoder and a post-processing net. It has as input, characters and using the Griffin-Lim algorithm which converts from the output, spectrogram frames.

The encoder receives as input a character sequences and extracts *robust sequential representations of text*[1]. To the input, a set of non-linear transformations are applied using what it is called a bottleneck layer with dropout. This improves generalisation and convergence of the method. The last step of the encoder is to transform the pre-net outputs into the final encoder output next used by the attention model. This is done by the 1-D convolution bank + highway network + bidirectional GRU (CBHG) module that reduces overfitting and mispronunciations when compared with a multi-layer RNN encoder.

The decoder used is a content-based tahn attention decoder that produces at each decoder time step the attention query with a stateful recurrent layer. The inputs of the decoder RNNs are the concatenation of the output of the attention RNN and the context vector. It was also used a stack of GRUs with vertical residual connections in order to speed up the convergence of the method.

For the seq2seq it was used as target a mel-scale spectrogram and to predict the decoder targets a fully-connected output layer.

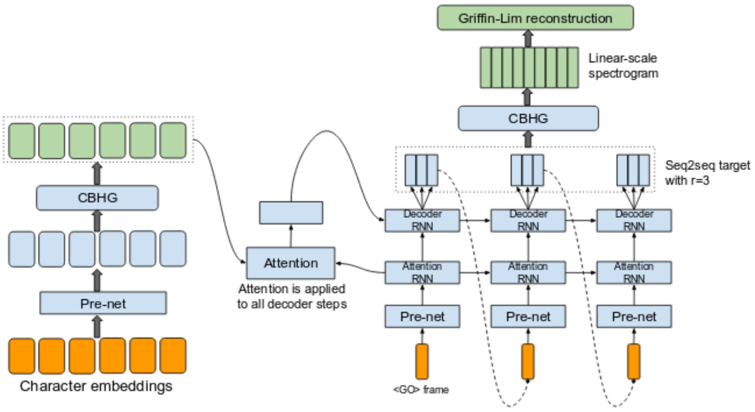


Figure 2.8: Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesise speech. [1]

[4] http://www.josesotelo.com/speechsynthesis/

As it can be seen in Figure 2.9, CBHG extracts representations from sequences and is constituted by a bank of 1-D convolutional filters, a highway network and a bidirectional GRU RNN. The purpose of the filters is to model local and contextual information.

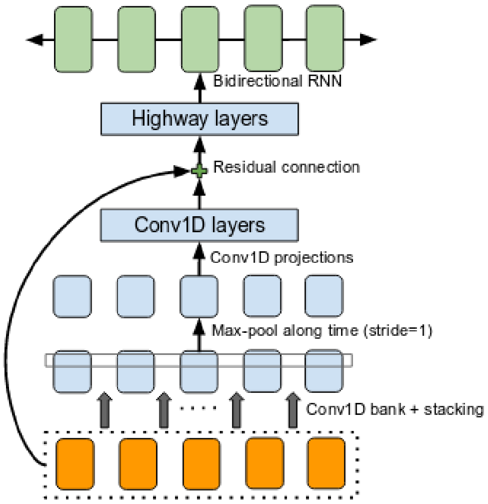


Figure 2.9: The CBHG (1-D convolution bank + highway network + bidirectional GRU) module adapted from Lee et al. (2016).[1]

In order to convert seq2seq target to waveform, it was used a post-processing network. The synthesiser used was Griffin-Lim [23] and implemented in TensorFlow[5] [24] and the post-processing net is a CBHG module in order to predict spectral magnitude sampled on linear-frequency scale.

Tacotron was trained to reach for 24,6 hours of speech data presented in an American English dataset.

Initially, the audio generated[6] was compared with already generated audio from different models. First, it was compared with audio from vanilla seq2seq model since the encoder-decoder model uses 2 layers of the residual RNN as Tacotron. An outcome was realised that the attention model in seq2seq holds in some frames, turning the signal synthesised bad in terms of intelligibility which destroys naturalness and duration. With Tacotron this was not a problem.

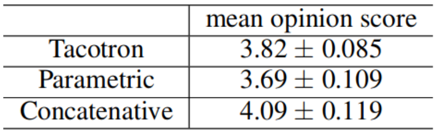
For the second comparison made was used a model with a modified encoder, from a CBHG to a residual GRU encoder with 2 layers. With this model could be observed more noise in the waves that may lead to mispronunciations when listening to the synthesised audio.

Finally, it was verified the importance of post-processing the net when more contextual information is used, this leads to better harmonics and high-frequency formant structure.

The MOS results obtained for this system are presented in Table 2.3. This test was made with natives, and they were asked to rate the naturalness of the audio generated. Each phrase got 8 evaluations, and 100 phrases were used. The results were at last compared with the results for a parametric based on LSTM and a concatenative system both currently in develop.

[5] https://github.com/Kyubyong/tacotron
[6] https://google.github.io/tacotron/

Table 2.3: 5-scale MOS evaluation. [1]



	mean opinion score
Tacotron	3.82 ± 0.085
Parametric	3.69 ± 0.109
Concatenative	4.09 ± 0.119

3

Expressiveness and Child Voice Synthesis

Contents

As referred previously, this chapter presents projects where expressive synthesis is used. Here we start by making a little overview of the theme.

Then, the first project presented was developed by AssistiveWare with the collaboration of Acapela Group and ExpressivePower^{TM} called Proloquo2Go[1]. This has the purpose of creating several expressive children's voice for application in tables to be used by Augmentative and Alternative Communications (AAC).

The second one is from Tainan University in Taiwan and the authors worked in order to develop a system to work with a Hidden Markov Model (HMM) more precisely a Multiple regression hidden semi-Markov model (MRHSMM) to synthesise expressive speech in a user friendly way for the user. This project focuses only on the system to produce emotions.[25]

Finally, we talk about Merlin applied to expressive speech synthesis.

Here, as well as in Chapter 2 the basic principals of the projects will be explained.

3.1 Overview

When talking about emotions there are three main approaches to do their modelling, the Categorical, the Dimensional and the appraisal-based approach. The categorical approach describes emotions in terms of discrete theories, stating the existence of a small set of basic emotions, that are universally recognised by our brains.[26] The conducted study for the categorical approach concluded that there are six basic emotions that can be recognised universally, happiness, sadness, surprise, fear, anger and disgust.

The dimensional approach that describes emotions as a space and uses a set of dimensions instead of discrete categories to measure them. These dimensions are: valence, arousal and potency. Valence indicates how pleasent or unpleasent a feeling is, or in other words how positive or negative. Arousal evaluates how active or passion/excited or apathetic a feeling is and finally, the potency dimension describes the sense of control over the feeling. [26] According to this approach, the affective states are not independent from one another; rather, they are related to one another in a systematic manner. [27]

The appraisal-based approach, is seen as an extension to the dimensional approach because it states that emotions are generated through continuous and subjective evaluation of both our own internal state and the state of the outside world. With this emotions is characterises with all the changes in all relevant components including cognition, motivation, physiological reactions, motor expressions, and feelings.[27]

3.2 Background Projects

3.2.1 Proloquo2Go

Proloquo2Go is a project developed by AssistiveWare with the collaboration of Acapela Group and ExpressivePower^{TM} that has the purpose of developing a Text to Speech (TTS) system that creates

[1] http://www.assistiveware.com/innovation/childrens-voices

genuine and not robotic children's voices for application in AAC.[1] This project synthesises speech using a database of recordings from a child. Since they created voices in several languages, American, British and Australian English, German, Swedish, Spanish and more recently French and Italian, a child was needed for each language apart from Spanish where a bilingual American Spanish-English child recorded the corpus. After recording, the raw data has to be processed and then repeatedly tested to ensure the naturalness and correct pronunciation of the words. For TTS synthesis with expressive voice, it has to be considered the different sound combinations for each word. With that knowledge, it is possible to build a corpus for recording with the minimum number of words that allow the maximum sound combinations possible. The purpose of the system is to create any necessary words with combining sounds from the recorded corpus.

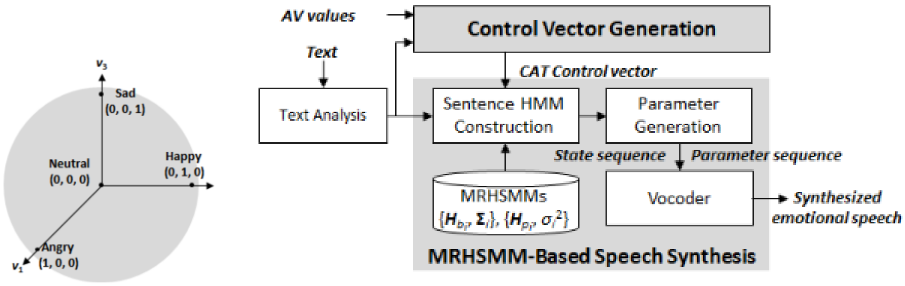
In order to have expressive voice for this project the recordings were extended for words and statements said in a more expressive way and for example animal sounds and these recordings have to be processed differently. It has to be considered the intonation of each word in order to develop special codes to define the speech characteristics. [2]

3.2.2 Generation of Emotion Control Vector

For this project, the authors studied a way to replace the vector-based method based on a control vector defined in the *Categorical (CAT) emotion space*.[25] With this control vector, it is difficult to be precise on choosing the emotion to synthesize. The authors tested the Arousal-valence (AV) space to an MRHSMM-based synthesis framework in order to define AV values in the AV space. To design the AV and the CAT emotion space it was used a Multidimensional scaling (MDS) method.

As it can be seen in Figure 3.1(a) is the proposed solution for the control vector generation. This has an MRHSMM-based speech synthesis that is trained with an expressive and emotional corpus and the respective transcriptions and control vectors in CAT emotion space as it is shown in Figure 3.1(b).

[2] http://www.assistiveware.com/innovation/expressivepower



(a) 3-dimensional CAT emotion space with four primitive emotions.

(b) Proposed control vector generation for MRHSMM based speech synthesis.

Figure 3.1: Proposed control vector generation for MRHSMM based speech synthesis. and CAT emotion space.

As depicted, the inputs are text and a vector with the AV values for the intended emotion in the space. Then the context-dependent labels are analysed by text analysis and with the use of the proposed control vector generation method, each phoneme control vector is transformed from AV space to CAT space.

As it can be seen in Figure 3.2, the AV space is a bipolar two-dimensional space where the horizontal dimension, valence, relates to emotions of pleasure or displeasure and the vertical dimension, arousal, relates to emotions of excitation-relaxation.

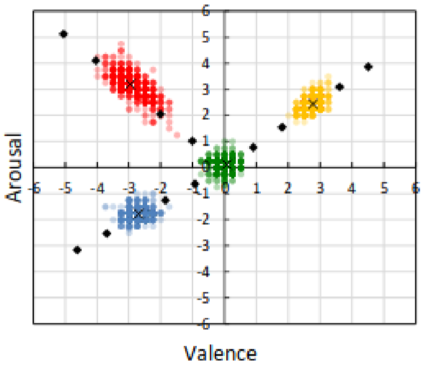
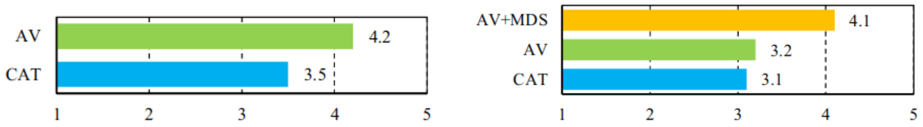


Figure 3.2: Emotion distributions in the AV emotion space. The yellow, red, blue, and green distributions denote the distributions of AV values of happy, angry, sad, and neutral, respectively. The black crosses represent the mean vectors for four emotions.

The HSMM is built from the context-dependent labels and the control vectors and then the system functioning is identical to a standard HMM-based speech synthesis system.

Finally, the output of the system is generated with the speech parameters generated by a parameter generation that considers dynamic features from the probability density function of the HSMM. These speech parameters are then passed through a filter-based vocoder, Mel log spectrum approximation (MLSA).

In order to evaluate the system, it was recorded approximately 4.000 utterances with about 1.000 utterances for each emotion, happy, angry, sad and neutral. In order to compare the obtained results regarding friendliness and emotion perception, such as for the methods mentioned in Chapter 2, the authors conducted the Mean Opinion Score (MOS) test. These results are presented in Figure 3.3.



(a) MOS results for friendliness test.

(b) MOS results for emotion perception test.

Figure 3.3: MOS results for friendliness and emotion.

3.2.3 Expressive Speech Synthesis in Merlin

This work is part of the Wikispeech project. [28] In this work the authors decided to use Merlin and make some changes in order to have a more expressive speech synthesis with this method. The right side of the figure 3.4 shows a modified architecture, designed to make the synthesised voice more expressive [29]. This solution aims at improving the realisation of prosody in TTS by advancing and implementing the model of prominence.

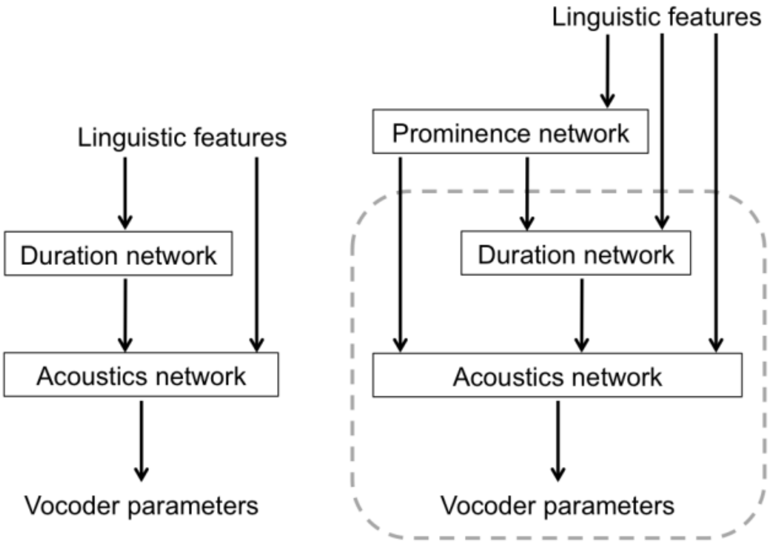


Figure 3.4: Left: standard Merlin DNN synthesis architecture. Right: modified architecture with explicit prominence modelling.

The modified method includes a third network to provide prominence features and to enable the control of prominence and emphasis. The main concerns in this project were related to the fact that prominence has a large number of acoustic features, such as duration, F0, energy, and spectrum. The results for this were considered to be very satisfying since they were able to control word prominence and the ratings for speech naturalness were better than for the baseline voice.

Although this method seems to be a first start to synthesise a more expressive speech, in this master thesis I did not work on this proposal.

4

Merlin Toolkit

Contents

In this chapter is described all the work and progress done with the Merlin Toolkit. Beginning with packages such as festival and festvox and libraries needed and the two methods used, slt_arctic and buid_your_own_voice. The practical work was done first in English in order to decide if it was worth it to repeat in European Portuguese such as it is purposed in the beginning of this thesis. All the work done in English with the method buid_your_own_voice, is described in section 4.1.1.

4.1 Familiarisation with Merlin Toolkit

4.1.1 Installing Merlin

Knowing all the existing methods for speech synthesis in use and development nowadays, we chose for this project the Merlin Method in the course of introduction to the research in Electrical and Computer Engineering. Mainly because it is a public domain[1] model and the only one with this condition at the time that the choice was made, can be trained with multiple languages and because of the previous experience of the team with the public domain Festival framework.[30]

For the IEEC curse the Merlin Method was installed as well as all the python libraries necessary. Initially for installing and testing the toolkit we followed a tutorial [2]. Merlin can be trained in two different ways, Demo and Full_Voice. The main difference between them is the number of utterances used, 50 and 1132 respectively. It is expected for each training to have a duration of 5 minutes if it is Demo or 1 to 2 hours if it is Full_voice, but this depends on the machine used and its characteristics.

Merlin toolkit runs in Linux so, first of all downloading and extracting from GitHub is done with a simple command in the terminal, this is referred in this report in the table 4.1, Step 1. Then, there are the requirements.txt to be installed that are refereed in Step 2 and it all has to be compiled with the command in Step 3 inside the directory `merlin/tools`.

Merlin runs over a python system so in order to run, we need several libraries [3]. Over the months of developing this thesis Merlin was changed and more libraries had to be installed in order to keep the system in development, updated. So the first commands for installing this libraries are indicated in Step 4 and 5 and they were installed in the beginning of this project.

Table 4.1: Steps in Merlin Toolkit

Step	Command
1	`git clone https://github.com/CSTR-Edinburgh/merlin.git`
2	`pip install - r requirements.txt`
3	`./compile_tools.sh`
4	`sudo apt-get install python-numpy python-scipy python-dev python-pip python-nose g++ libopenblas-dev git libc6-dev-i386 csh`
5	`pip install numpy scipy matplotlib lxml theano bandmat`
6	sudo apt-get install cmake
7	pip install pysoundfile

Nowadays, in order to install Merlin, step 6 is necessary to have installed before every other step presented in the table 4.1. Merlin has also been increased, so there are several ways to synthesise

[1] https://github.com/CSTR-Edinburgh/merlin

[2] http://jrmeyer.github.io/merlin/2017/02/14/Installing-Merlin.html

[3] https://github.com/CSTR-Edinburgh/merlin/blob/master/INSTALL

voice. The first one tested was slt_arctic/s1 in the IIEEC curse. Then it appeared a new method for the slt_arctic, s2 and for this one the Step 7 is required. This first two ways of synthesis allow us to familiarise with the merlin method but with this, the phrases synthesised are selected from the ones used in the training.

In order to synthesise our own voice or at least different sentences with different characteristics and other training datasets the way to run Merlin is with the method build_your_own_voice/s1. This method has several steps, also presented in the slt_arctic but with the possibility to change every parameter. This steps are presented in the table 4.4 and will be explained further in the report.

Table 4.2: Build_your_own_voice steps

Step	Command
1	01_setup.sh
2	02_prepare_labels.sh
3	03_prepare_acoustic_features.sh
4	04_prepare_conf_files.sh
5	05_train_duration_model.sh
6	06_train_acoustic_model.sh
7	07_run_merlin.sh

Build_your_own_voice method requires speech tools[4], festival[5], festvoz[6] and HTK[7] to be installed in this order. The steps to install each of this packages are presented in table 4.5. It is also necessary to have in mind the libraries needed to install each of this packages. And htk was developed to work in a system with 32 bits so it is needed to be careful with the systems bits of the used computer.

Table 4.3: Build_your_own_voice steps

Step	Command
1	01_setup.sh
2	02_prepare_labels.sh
3	03_prepare_acoustic_features.sh
4	04_prepare_conf_files.sh
5	05_train_duration_model.sh
6	06_train_acoustic_model.sh
7	07_run_merlin.sh

4.1.2 Training Merlin with slt_arctic

The databases used to make the experimental part belong to CMU ARCTIC which were specially designed for research in speech synthesis and are publicly available. The one used specifically is called slt_arctic and like the other dadasets that belong to the CMU their utterances are 16 bit, mono waveforms and sampled at 16kHz.

In order to understand Merlin, we checked several code files in the beginning of these work. One of the conclusions taken from this investigation relates to the fact that Merlin slt_arctic toolkit has 5 steps and is used only to understand the potentialities of merlin Toolkit. Step 1 creates a file `global_settings.cfg` which has data about the configuration files. A .zip with the audio samples to

[4] https://github.com/festvox/speech_tools
[5] https://github.com/festvox/festival
[6] https://github.com/festvox/festvox
[7] https://github.com/ibillxia/htk_3_4_1

use and the information about the configuration, is downloaded from which are created configuration files for the duration and the acoustic models, `duration_demo.conf` and `acoustic_demo.conf`. This .zip has 3 directories, `lab`, `wav` and `merlin_baselin_practice`. The first two have the training files, .wav and .lab necessary for training the net. The last one contains other 3 directories, `test_data`, which contains the file describing what is to synthesise, `duration_data` that has information about the labels and state and phone alignment. The last directory is `acoustic_data` which has the necessary files and information to train the acoustic model.

The `merlin_synthesis.sh` file calls a script, `prepare_labels_from_txt.sh` which uses the festival front-end that is placed in the directory misc to extract the corespondent features from the dataset used. Step 2 and 3 consist in training duration and acoustic models of the Neural Network (NN) respectively. Step 4 is wave file synthesis that in these demo versions consists in the re-synthesis of some dataset files.

In order to synthesise new speech sentences it is written on the instructions for Merlin slt_arctic thar we are supposed to run the full voice and then run the command: `./merlin_synthesis.sh`. However, for the command to work we have to give a the sentence to synthesise or a file .txt containing the set of sentences we want. The command is actually in the format `./merlin\_synthesis.sh sentence.txt`.

4.2 Build_your_own_voice in Merlin toolkit

In order to synthesise a different voice we started to work with the Build_your_own_voice in Merlin toolkit, first with the dataset from CMU ARCTIC already used and then with the LJSpeech [8] [31]. The LJSpeech dataset is public domain, has 13.100 short audio files which correspond to a total of approximately 24 hours of speech. Another characteristic from this dataset is that the audio files have a sample rate of 22050Hz and because of that we studied which sampling frequencies are accepted from Merlin. So we can only use files with sampling frequencies of 16kHz or 48kHz, (this last one is very uncommon). In order to use this second dataset, we had to convert the sampling frequency from each file to a frequency accepted from Merlin.

Table 4.4: Code used for sampling wav files

Steps	Shell code
1	`cd wav`
2	`mkdir out`
3	$forfin./*.wav; dosox"\$f" - r16000./out/\$f; done$
4	$sox - -i - rfile.wav$

Although the LJSpeech has 13,100 audio files we used a maximum of 6,119 files as we can see in table 4.6. These files were chosen according to the most relevant features and utterances to have in the dataset. This choice was made mainly because of the performance of the machine where this system was tested. Another reason for this choice is that 6119 audio files is already a very big

[8] https://keithito.com/LJ-Speech-Dataset/

dataset when we compare it to the CMU ARCTIC. Besides, with the results from this dataset size and comparing them with the results obtained with trainings made to debug with smaller datasets from the same database were very consistent, allowing us to draw conclusions about the results with this database.

Table 4.5: Build_your_own_voice steps

Step	Commands and Steps
1	cd merlin/egs/build_your_own_voice/s1
2	mkdir database
3	copy /wav/ and /txt/ of the dataset to /database/
3	./01_setup.sh <voice_name>
4	Change global_settings.cfg in folder /conf/
5	./02_prepare_labels.sh <path_to_wav_dir> <path_to_text_dir> <path_to_labels_dir>
6	./03_prepare_acoustic_features.sh <path_to_wav_dir> <path_to_feat_dir>
7	./04_prepare_conf_files.sh <path_to_global_conf_file>
8	./05_train_duration_model.sh <path_to_duration_conf_file>
9	./06_train_acoustic_model.sh <path_to_acoustic_conf_file>
10	./07_run_merlin.sh <path_to_text_dir> <path_to_test_dur_conf_file> <path_to_test_synth_conf_file>

In order to train Merlin with this datasets there are seven steps necessary to follow. In the first one we have to make some changes in order to accomplish our purposes. So in this first step we change the file to include the correct path for the required tools, festival, festvox and HTK. We change or include also the sentences that we want to synthesise. And we also include the question_file that we want to use. This file contains the phonological characteristics from the language to use. For the English synthesis, this parameter was not changed because we had already a question_file for English in Merlin. Besides that we have to chose the number of files to train, validate and test Merlin. This is chosen with a percentage from the the total number of files, for the training we use 90% of the files and for the validation and test we user 5% of the files for each. As we can see in table 4.6 this were the values used in order to train Merlin with the English Voice.[9] The other 6 steps don't require any special action, we just have to run them in the correct order and with the correct inputs.

Table 4.6: Number of files in global_settings_cfg for 6119 utterances

Total number of files	6,119
Training Files	5,519
Validation Files	300
Test Files	300

To understand how the duration of dataset relates with the duration of training, we started by count the duration of each dataset we choose to work with. The commands used are presented in table 4.7. In the first possible proposed way, we have to convert the $.wav$ files in $.mp3$, step 1.1, and then with step 1.2 we sum up the duration of each file. However, with this two steps we don't count with the header size from each file. So we used step 2 that does not demands the files to be $.mp3$ and that thanks into account the headers.

[9] https://github.com/CSTR-Edinburgh/merlin/issues/203

Table 4.7: Steps to count dataset duration

Step	Command
1.1	$for\ i\ in\ *.wav;\ do\ lame\ -b\ 320\ -h\ "\$\ \{i\}"\ "\$\{i\%.wav\}.mp3";\ done$
1.2	$for\ file\ in\ *.mp3;\ do\ mp3info\ -p\ "\%S\ \backslash n"\ \$\ file;\ done\ \mid\ paste\ -sd+\ \mid\ bc\ \mid\ awk\ '\{print\ strftime("\%H:\%M:\%S",\ \$0)\}'$
2	$ls\ -als\ \mid\ awk\ 'BEGIN\{sum=0\}\{sum=sum+\$6\}END\ \{print(sum-7303*44)/(2*16000*3600)\}'$

In order to understand the duration of the trainings for Merlin, we did several time measures in some of the trainings we did and related them with the duration of the dataset used. These results are presented in table .

Table 4.8: Datasets used and training duration

Dataset	Utterances	Dataset Duration	Training Time
slt_arctic_demo	50	00:20:32	00:50:32
slt_arctic_full	1,132	00:50:32	04:20:24
LJ Speech [31]	60	01:06:47	00:38:23
	1,000	02:53:36	12:51:04
	2,000	04:47:03	25:41:47

From table 4.8 we can conclude that the trainig duration does not only depend on the duration of the dataset but also depends on the number of files.

4.2.1 Results with the English Voices

In order to conclude about the quality of the results obtained we chose to synthesised some phrases that we present below. For the classification of this obtained results, we just listened to the audio and observed the waveforms of the files in wavesurfer.

- "Hello, I am Catarina."
- "Hi, this is a demo voice from Merlin."
- "Printing, in the only sense with which we are at present concerned, differs from most if not from all the arts and crafts represented in the Exhibition in being comparatively modern."

Some of the conclusions taken from these results were relative to the LJSpeech dataset. This one contains a more expressive speech in terms of energy of speech, but Merlin works with probabilistic methods that began to calculate the means of the energies of each syllable with the purpose to define the pitch of the speech. So, when we have a dataset with energy changes, we can detect some noise in the audio produced and because of that we can also conclude that some of the speech created with the slt_arctic_full has a better quality than with the LJSpeech, even if the last dataset is bigger.

With these observations and knowing how to create an expressive speech synthesizer with Merlin, the next steps were directed to the synthesis of a Portuguese voice where the dataset used had to be most neutral in terms of energy .

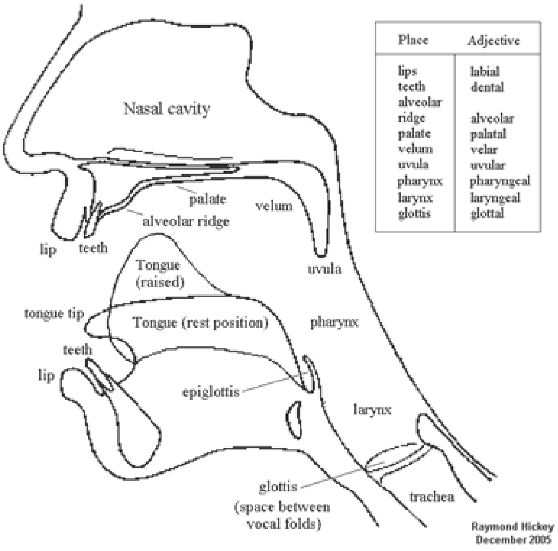
4.3 Synthesis for European Portuguese

In order to synthesise the Portuguese voice it was necessary to make changes in Festival and to create a proper question_file for the Portuguese phonology. For Festival, the changes were mainly done by changing the English voice for the Portuguese one l2f_sgpAlign_diphone. That change was made by adding to festival the directories from the Portuguese voice and by changing the file $init.scm$ which is the file responsible for the initialisation of Festival. The command that has to be changed in this file is: $(voice\_l2f\_sgpAlign\_diphone)$.

4.3.1 European Portuguese Phonology

European Portuguese is language composed phonetically by 37 phonemes, including the 14 vowels, 9 diphthongs and 23 consonants. We used the SAMPA (Speech Assessment Methods of Alphabet) script to understand the articulation classifiers pre-established.

This phonemes can be classified from the speech articulation system by manner which can be labial, coronal, alveolar, dental, velar and palatal/dorsal. Also, by place that can be nasal, lateral and trill that can also be considered liquid, plosive and fricative that can be voiced and unvoiced. Also by place, we can have semi-vowels that are defined as *phonetically similar to a vowel sound but with function as the syllable boundary*. It can be observed in the figure 4.1 the several organs that form the speech system and some of the correspondent manner and place classifiers for consonants.



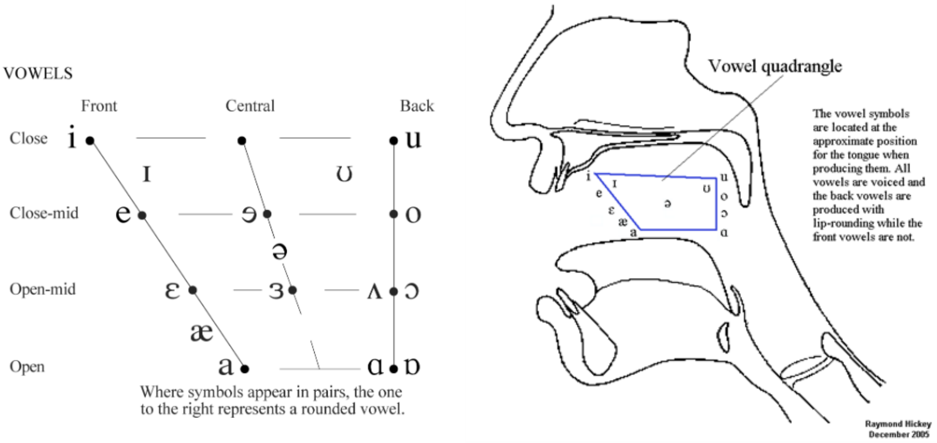
Place	Adjective
lips	labial
teeth	dental
alveolar ridge	alveolar
palate	palatal
velum	velar
uvula	uvular
pharynx	pharyngeal
larynx	laryngeal
glottis	glottal

Figure 4.1: Speech Organs. [10]

Table 4.9: European Portuguese Consonants

Consonants							
		Labial	Coronal	Avelar	Dental	Velar	Palatal/Dorsal
Plosive/Occlusive	Voiced	b	d			g	
	Unvoiced	p	t			k	
Fricatives	Voiced	v			z		Z
	Unvoiced	f			s		S
Nasal		m		n			J
Lateral/Liquid				l/l~			L
Trill/Liquid			r			R	
Semi-vowels		w w~					j j~

As consonants, vowels can as well be classified from the tongue position. The classifiers are then, regarding the place of the mouth, front, central and back, the opening of the mouth, close, close-mid, mid, open-mid, and open, and regarding the produced sound, oral and nasal. It can be seen in figure 4.2(a) we have a vowel quadrangle and then in figure 4.2(b) this quadrangle placed in the mouth.[11]


VOWELS
Front
Central
Back
Close
Close-mid
Open-mid
Open
i
ɪ
u
ʊ
e
ɘ
o
ə
ɛ
ɜ
ʌ
ɔ
æ
a
ɑ
ɒ
Where symbols appear in pairs, the one to the right represents a rounded vowel.
Vowel quadrangle
The vowel symbols are located at the approximate position for the tongue when producing them. All vowels are voiced and the back vowels are produced with lip-rounding while the front vowels are not.
Raymond Hickey December 2005

(a) Vowel Quadrangle **(b)** Vowel Quadrangle placed in the Mouth

Figure 4.2: Vowel Quadrangle

Table 4.10: European Portuguese Vowels

Vowels						
	Oral			Nasals		
	Back	Central	Front	Back	Central	Front
Close	i		u	i~		u~
Close-mid	e		o	e~		o~
Mid			@			
Open-mid	E	6	O		6~	
Open		a				

4.3.2 Steps for synthesis

The modifications made in question_file were based in Portuguese phonology that is described in section 4.3.1 and the necessary Festival files that contain information about the syntax used by it.

[11] http://www.aston.ac.uk/lss/research/lss-research/ccisc/discourse-and-culture/west-midlands-english-speech-and-society/sounds-of-english/sound-production/

However we used the typical phonology for European Portuguese and based our system in Speech Assessment Methods of Alphabet (SAMPA) we had to change the character defined by "6" and "$\sim$" since these were characters already used by HTK for another function. So, we replaced this characters with "A" and "y" respectively.

After having the question_file, we repeated all the steps already done for the English. We chose the first 60 utterances from the *Vitalina* dataset in order to test and debug Merlin. The only change relative to the English synthesis relates to the encoding of the $.txt$ files that will be defined as $UTF-8$ but Merlin only works with lower codifications such as $ISO-8859-1$. So we had to do the step 6 in the directory indicated by step 5 in table 4.11.

Finally, in order to optimise the process of synthesising speech in Merlin, we created a $script.sh$ that can do all the steps from Merlin from one to seven that corresponds in the table 4.11 to steps 4 to 13. However, for this to work we have to change the script from step 4 in table 4.11 in order to produce the exact $global\_settings.cfg$ file to run merlin. So the script does not do steps 5, 6 a 7 of the table.

Table 4.11: Build_your_own_voice steps For European Portuguese

Step	Commands and Steps
1	cd merlin/egs/build_your_own_voice/s1
2	mkdir database
3	copy /wav/ and /txt/ from *Vitalina* to /database/
4	./01_setup.sh *Vitalina*
5	cd experiments/vitalina/test_synthesis/txt
6	recode UTF-8..ISO-8859-1 *.txt
7	Change global_settings.cfg in folder /conf/
8	./02_prepare_labels.sh <path_to_wav_dir> <path_to_text_dir> <path_to_labels_dir>
9	./03_prepare_acoustic_features.sh <path_to_wav_dir> <path_to_feat_dir>
10	./04_prepare_conf_files.sh <path_to_global_conf_file>
11	./05_train_duration_model.sh <path_to_duration_conf_file>
12	./06_train_acoustic_model.sh <path_to_acoustic_conf_file>
13	./07_run_merlin.sh <path_to_text_dir> <path_to_test_dur_conf_file> <path_to_test_synth_conf_file>

After debugging the system, the final Merlin synthesizer was trained with the 7,303 audio files from Vitalina. This dataset from VoiceInteraction was chosen because it has recordings from a female voice with higher pitch, thus closer to a child's voice than any other possible choice of adult voices.

Such as we did for the English voice and explained in section 4.2, the choice of the number of training, validation and testing files is presented in table 4.12.

Table 4.12: Number of files in global_settings_cfg for *Vitalina*

Total number of files	7,308
Training Files	6,578
Validation Files	365
Test Files	365

In order to understand the duration of the trainings for Merlin, we did several time measures in some of the trainings we did and related them with the duration of the dataset used. These results are presented in table 4.13.

Table 4.13: Datasets used and training duration

Dataset	Utterances	Dataset Duration [hh:mm:ss]	Training Time [hh:mm:ss]
Vitalina [31]	60	00:20:45	00:50:32
	1,100	01:18:36	09:17:28
	7,303	15:32:11	72:04:08

We also measured the necessary time to synthesise different amount of Voice in Merlin with 1,100 and 7,303 training utterances. The obtained results are presented in table 4.14. From this we can conclude that, as expected, the amount of synthesised phrases and the size of the training dataset does not have a big influence in the duration of synthesis. The conclusion that can be taken from technical experience regarding the time of synthesis is that this also depends on the computer load at the moment of synthesis. If is using RAM for another process it can take a little bit more time to synthesise the phrases. However, for me, it never took more than 2 minutes to synthesise phrases in Merlin.

Table 4.14: Number of files in global_settings_cfg for Vitalina

Number of training files	Number of synthesised files	Synthesis duration [mm:ss]
1,100	1	01:40
	32	02:57
7,303	1	01:56
	32	02:42

4.3.3 Synthesised Corpus

All of the synthesised sentences that were used to evaluate Merlin can be found below. We decided to have four main groups of synthesised phrases. The first one relates to common words and sentences and to this group belong the first four phrases. The next four phrases belong to Portuguese news and to the second group of phrases. Then we present the eight phrases that belong to the third group and relate to meteorology. Finally, we present the phrases that belong to the fourth and final group which are phrases from children books. In total we synthesised thirty phrases in order to take conclusions about the synthesis quality of Merlin for European Portuguese. The results are presented in chapter 5.

- Estou a trabalhar com o Merlim para sintetizar voz em Português para a minha dissertação de mestrado.
- Eu almoço um bom almoço.
- Vou fazer um bolo. Pronto, já só precisa de ser metido no forno.
- E eu tomei o chá com a minha avó, enquanto esta me contava uma história.
- O memorando sobre a recuperação do material de guerra roubado em Tancos não deixou qualquer rasto no Ministério da Defesa. Ou seja, pura e simplesmente não existe.
- Não é uma escolha, eu sou assim!

- A selecção portuguesa defronta a sua congénere polaca para a Liga das Nações. Um triunfo deixa as portas abertas para a final a quatro da competição.

- Quando planeados e ordenados, segundo critérios científicos, os espaços verdes podem vir a melhorar o ar das cidades ao reduzir a concentração de poluentes e ter uma melhor eficiência em termos de custo — benefício do que as medidas tecnológicas.

- A tempestade tropical Michael está prestes a transformar-se em furacão, ameaçando com fortes chuvas o oeste de Cuba.[12]

- As temperaturas podem chegar aos 22 graus no Alentejo e aos 19 no Algarve, neste domingo de Páscoa.

- A chuva só deverá regressar a Portugal continental na segunda-feira.

- A previsão aponta também vento fraco, soprando temporariamente moderado, de nordeste nas terras altas do Norte e Centro até ao início da manhã e do quadrante oeste durante a tarde nas regiões Centro e Sul.

- Céu muito nublado, diminuindo de nebulosidade a partir do início da tarde.[13]

- Períodos de chuva, passando a aguaceiros fracos a partir da madrugada até ao fim da manhã.

- Para quinta-feira e sexta-feira está previsto céu limpo, com as temperaturas na ordem daquilo que tem sido nos últimos dias.

- No Arquipélago dos Açores, para quinta e sexta-feira prevê-se que continue a chover e a trovejar como tem sido até agora em Ponta Delgada, mas com elevado grau de abrandamento, já que hoje e amanhã serão os períodos mais críticos da pluviosidade e da trovoada nas ilhas açorianas.

- Como ali havia horas para tudo e regras para tudo os dias pareciam todos iguais.

- Só depois se punha de pé sempre o último da fila lá ia pelo corredor arrastando-se penosamente e nem sequer tinha o cuidado de disfarçar longos bocejos.

- O pão quente e o leite morno exalavam um cheiro de fazer crescer água na boca.

- Estamos no Inverno e lá fora está muito frio.[32]

- A casa do Vítor e do Tim está coberta por uma grossa camada de neve, porque nevou durante toda a noite!

- Ele afasta as cortinas e olha pela janela. Iúpi!, grita ele muito contente. Anda ver Vítor! Nevou toda a noite e está tudo branquinho lá fora. Vamos brincar na neve?[32]

[12] https://www.tempo.pt/noticias/previsao/outono-tarda-em-chegar-tempo-esta-semana.html

[13] http://www.ipma.pt/pt/index.html

- O cortejo pára. Faz uma pausa. As crianças rodeiam os músicos e as majoretes: Que instrumento esquisito! O que é? É um trombone. É perigoso? pergunta o Pantufa. Claro que não. Tens cada pergunta![33]

- O violino é melhor ou talvez o violoncelo?... Gostas assim tanto de música? pergunta a monitora das majoretes. A minha prima Isabel é violoncelista. Toca no concerto que vai começar daqui a bocado. Tenho dois convites.[33]

- No intervalo Anita é apresentada a Isabel. Como se chega a violoncelista? Vem terça-feira a minha casa que eu explico-te. Temos tempo. Estamos de férias. Adorava, diz Anita. Não te esqueças de avisar os teus pais... Está bem? Está bem. Eu falo com eles.[33]

- Vamos começar por fazer uma experiência, para ver se tens bom ouvido.[33]

- Como é que ele pode saber quem eu sou se nunca me tinha visto? Aproxima-te, para eu te ver melhor. Proíbo-te que bocejes! Posso-me sentar? E quando é isso? Mas para que é que isso te interessa?[34]

- Mas para que é que eu quero um elefante dentro de uma jibóia? As jibóias são muito perigosas e os elefantes muito incomodativos. O meu sítio é muito pequenino... Preciso é de uma ovelha. Desenha-me uma ovelha. [34]

- Mas o que vem a ser aquela coisa? Aquilo não é uma coisa. É o meu avião. O quê?! Tu caíste do céu?! Caí. Ah! Que engraçado![34]

- Com as suas grandes patas, o Simão começa a saltar. E os gafanhotos verdes também o querem imitar.[35]

4.4 Corpus for Synthesis with a Child Voice

The development of the corpus must take into account the age of the target speaker. We considered as a good target children with 7-8 years.

Although this part of our research was barely started, we have investigated some vocabulary sources. The target vocabulary will naturally be very short for children aged 4 or less, and it can be easily be structured into semantic categories such as family, animals, fruits, vegetables, and feelings. For older children, this categorisation may be less relevant. A child with 6 years in Portugal is already expected to know at least 10.000 words [36].

Because prompting and reinforcement sentences will be naturally important in the context of a child-robot interaction, we have also investigated the list of recorded items used in the INSIDE project that was kindly lent for this thesis. This is used for interactive games with the robot and the children with Autism Spectrum Disorder in order to stimulate them to communicate.

There are three different game proposals in the INSIDE project, the first is finding balls hidden in a room, the second is to make a puzzle and at last *tangram* that is similar to a puzzle, each piece has a shape and with a set of pieces they can make animals or other designs. The robot has speech for greetings, invitation to play, asking for help and asking to help others. It can also ask to show and invite to see a movie and make questions and comments about it and say goodbye in different ways.

5

Experimental Results

In order to evaluate the results obtained, we created a form based in MOS comparing our results with the in house DIXI system.[1]

Contrarily to the initial expectations, the DIXI voice corresponding to Vitalina was not available, hence we chose *Violeta*, another female European Portuguese voice in DIXI. This enabled a first comparison between Merlin using Vitalina, with different dataset sizes.

A second comparison was made at a later stage by training Merlin with a dataset of 1,000 phrases from Violeta, thus enabling a more fair comparison with the same voice in both systems.

In the first test, the participants were asked for a subjective MOS score (see 1.1 more specifically in table 1.1). The results are presented in the figure 5.1 and in table 5.1 where we compare the results with the dataset duration. We also chose to compare the MOS obtained inside each group of phrases. We can divide our results in four groups, the first with common phrases, the second with news, the third with phrases from a limited domain (meteorology) and finally the fourth with child's book phrases with dialogues. These results are presented in figure 5.2.

Table 5.1: MOS obtained for each method

Method	Duration	MOS
Merlin_Vitalina_1,100	1	3,73
Merlin_Vitalina_7,300	14	3,72
Merlin_Violeta_1,000	1 hour	2,95
DIXI_Violeta	6 hours	2,87

,

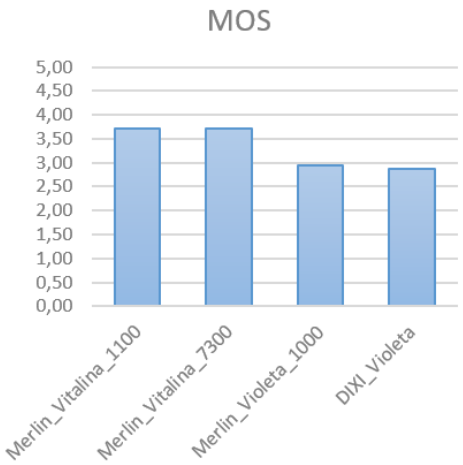


Figure 5.1: Graphic with MOS for each method

As we can observe, the higher MOS has been obtained by the Merlin synthesizer, trained with the Vitalina dataset with 1,100 phrases which consists in approximately 1 hour and 30 minutes in the dataset. The results obtained by training in Merlin with 1,100 and 7,300 phrases are very close in terms of MOS scores. In general, Merlin with 1,100 phrases is slightly better, but if we take into account the MOS distribution per group of synthesised phrases in figure 5.2 there are some situations where Merlin trained with 7,300 phrases from Vitalina is a better method.

[1] https://www.l2f.inesc-id.pt/demos/voices/

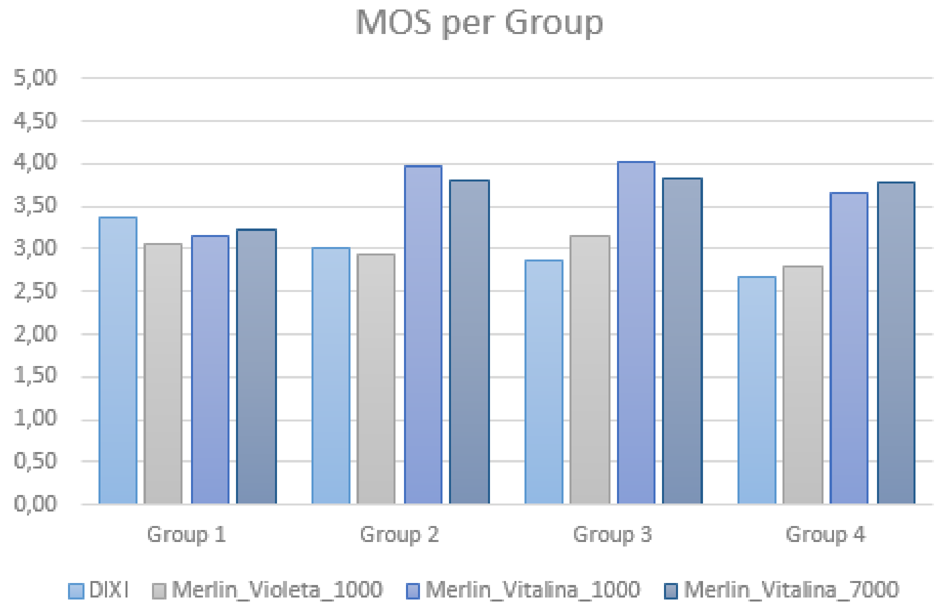
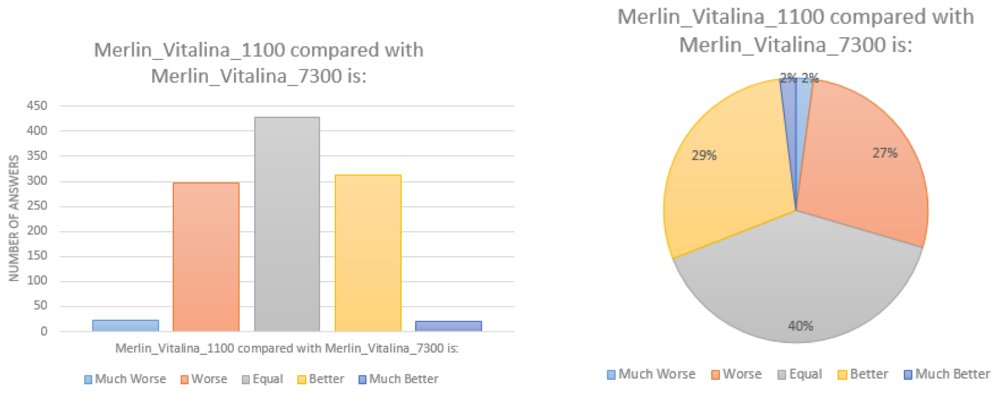


Figure 5.2: MOS obtained per group of phrases

In the second part of the tests, we synthesised 30 phrases for each of the described synthesis methods and voices. So we made groups of four phrases, one from each method and we asked for the comparison between the phases within the same group. For each phrase compared with the others three in the group and we asked in the form to characterise each phrase comparison as much better, better, equal, worse or much worse. We repeated this processes making all the possible combination between each group of synthesised audio. In total we had 30 parts in the form for each group of four phrases. The results of this form are presented in figures 5.6, 5.4 and 5.5.



(a) Merlin_Vitalina_1100 compared with Merlin_Vitalina_7300

(b) Merlin_Vitalina_1100 compared with Merlin_Vitalina_7300 in percentage

Figure 5.3: Merlin_Vitalina_1100 compared with Merlin_Vitalina_7300

As we can observe in figures 5.3(a) and 5.3(b), the two different synthesis used in Merlin are very similar. The results show that 40% of the inquired people found both synthesis equal, 29% found Merlin with 1,100 phrases to be better than when trained with 7,300 phrases and 27% of the inquired said that is worse. From these values we could conclude that Merlin with 1,100 phrases in training is a better method since we have 40% of equal classifications, 29% of better classifications and 2% of much better classifications. These values give us a total of 71% of the answers. This means that 71% of the inquired people found Merlin with 1,100 phrases of training is better or equal than Merlin with 7,300 phrases. However this has a very little interval compared with the 69% of the results saying that Merlin with 1,100 phrases of training is worse or equal than Merlin with 7,300 phrases.

In order to have better conclusions about this results, as we explained above while explaining the MOS evaluation, we have 4 main group of phrases. This groups are presented in figure 5.4 so we can understand where each training in Merlin has a better result comparing the different synthesis that were made with *Vitalina*. As it can be seen, in group 3, Merlin with 1,100 phrases behaves better than Merlin with 7,300 phrases. However in the other 3 groups the results are not very conclusive as we realised above with the MOS test in figures 5.1 and 5.2 and with the figure 5.6 were we saw the general results for this comparison.

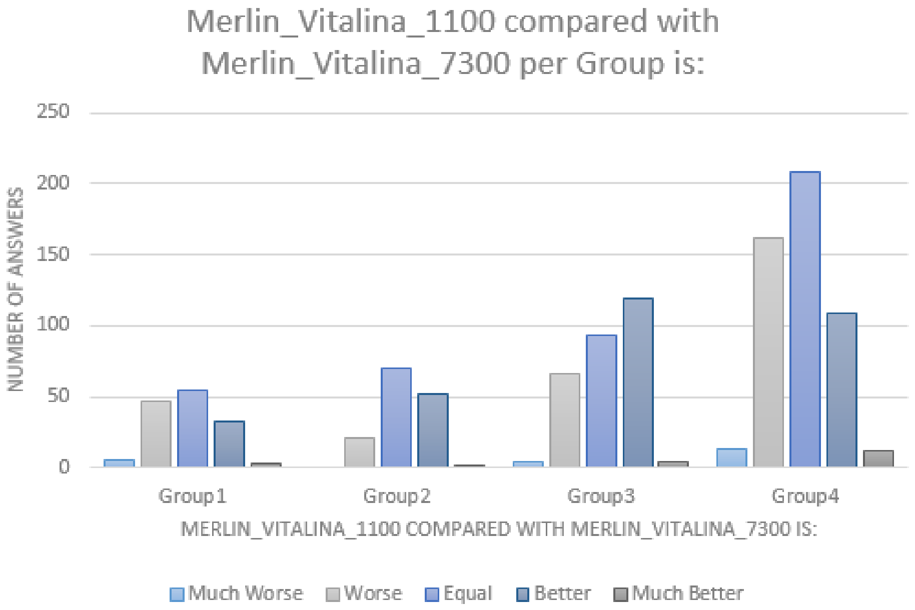


Figure 5.4: Comparison between the *Vitalina* in Merlin trained with 1,100 and 7,300 phrases

Another important comparison we have to make is with DIXI, the concatenive speech synthesis Method. As it can be seen in figure 5.5 are presented the results corespondent to the comparison of the other methods with regard to DIXI. From this, we can conclude that clearly Merlin is a better method to synthesise speech when trained with *Vitalina* than DIXI or even Merlin trained with *Violeta*.

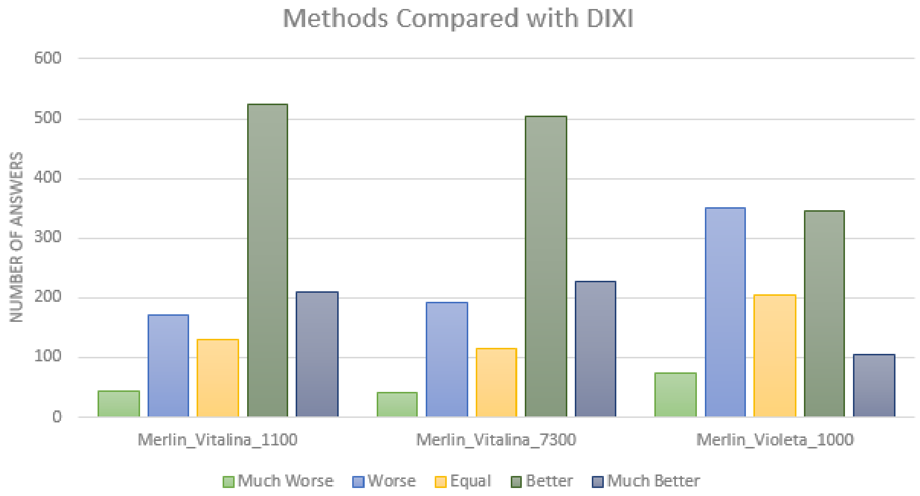
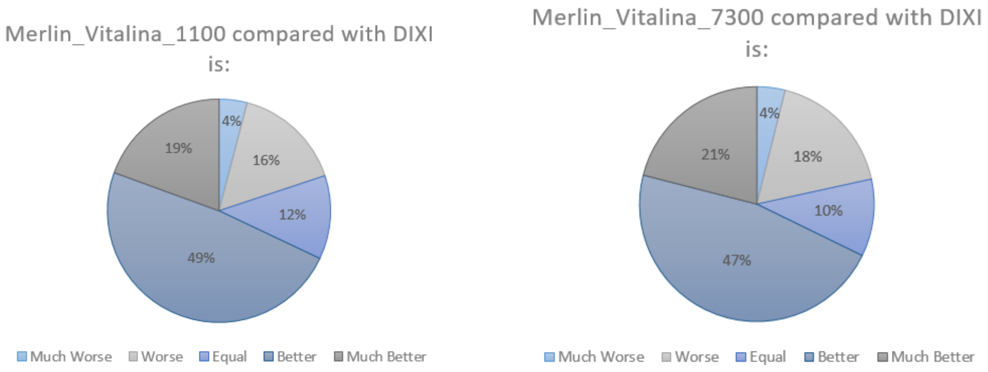


Figure 5.5: Comparison between the *Vitalina* in Merlin trained with 7,000 and *Violeta* in DIXI

From figure 5.6(a) the results show that 49% of the inquired people found Merlin with 1,100 phrases to be better than DIXI and 19% of the inquired said that is much better. From this values we could conclude that Merlin with 1,100 phrases in training is a better method since we have a total of 68% of the inquired people giving a higher than better classifications and that 12% of the people found them to be equal.



(a) Merlin_Vitalina_1100 compared with DIXI_Violeta

(b) Merlin_Vitalina_7300 compared with DIXI_Violeta

Figure 5.6: Merlin_Vitalina_1100 and Merlin_Vitalina_7300 compared with DIXI_Violeta

6

Future Work

For this master thesis there are two main steps to improve in the future.

The first one is related to recording a child to synthesise a child voice with Merlin. Merlin is prepared to synthesise any wanted voice with the only requirement of providing a new voice recording. In order to synthesise the child's voice it is only needed to the recordings of the child. This requires several investigations about how to proceed to record the child and the corpus to record. The corpus has to be thought about since from the results that we obtained when comparing Merlin_Vitalina_1100 with Merlin_Violeta_1000 we could conclude that with *Violeta* the synthesis in worse. Another thing that has to be taken into account is the minimum time of the recordings. The best results were obtained with approximately 1:30 hours of recording for training. With this improvement we can train Merlin with the so much wanted child voice to use with the robots from the Monarch and INSIDE projects.

Finally, the other possible improvement is to modify Merlin in order to improve speech expressiveness. During this work, we studied a method that accomplished to include a model of prominence to improve the naturalness of speech. Our first suggestion to accomplish this improvement is to start working with this proposed method and to modify the model that was used for the American English in order to work for the European Portuguese.

The second suggestion is to change methods replacing Merlin with Tacotron for example. This is a method that is open source and already synthesises speech with very good naturalness results. The problem with this second suggestion is that the system would need to be changed in order to synthesise speech in European Portuguese. Another concern is related to the fact that Tacotron does not uses Festival.

7

Conclusions

This report describes the work developed for the Master Thesis in in Electrical and Computer Engineering. We present an overview of the main theme, speech synthesis, and the motivation that led to the proposed work. Since the main task of this thesis is to synthesise an European Portuguese voice, I made some research about the phonological characteristics of the Portuguese speech. Also, the chosen method to synthesise Portuguese speech was Merlin, mainly because it has an open source code and uses Festival as frontend which was already used for DIXI, a concatenative speech synthesis system.

As it was described in this paper I managed to synthesise an European Portuguese voice with two different datasets. For one of those datasets, I did several synthesis using a different number of phrases from them. The main technical problem that I encountered during this project relates to the machine characteristics necessary to run Merlin without problems. Therefore, with a computer with 2Gb of RAM memory and an Intel Core 2 Duo CPU, the results were MemoryError in the acoustic model training and with more tests we realised that 4Gb of RAM were not enough as well. Finally, with a computer with 6Gb of RAM memory and an Intel Core i7 CPU and system Lubuntu 17.04, 64 bits version we got to run Merlin without any major issue, but with very large times of training

Given the results presented in section 4, we can conclude that the main purpose was achieved, since the results comparing Merlin with DIXI showed that the majority of the inquired people preferred Merlin trained with Vitalina over DIXI. We could also observe that when comparing Merlin synthesis with Violeta and DIXI the results were inconclusive. These results can be justified with the amount of phrases from Violeta used to trained Merlin. However, since the dataset is just 100 phrases smaller than the one from Vitalina used, the results can be explained if the corpus of these two voices is different, since Violeta can be lacking some important phoneme sequences. Thus, we believe that the main goal of this Thesis was achieved and it can also be recognised the importance of these work for future applications including maybe the replacement of DIXI.

For future work there are some possible directions this research can take. The first one is to record a Child in order to synthesise a child's voice that can be applied in the robots used for the Monarch and the INSIDE projects in order to have interaction with the children with a child's voice. This may increase the help given to these children reducing the apprehension felt by them since the communications are done through a voice from an equal. A more challenging future possible application is to develop a method to have a more expressive voice in Merlin. This could be achieved for example using the method of adding a prominence model. [29]

Bibliography

[1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," *CoRR*, vol. abs/1703.10135, 2017. [Online]. Available: http://arxiv.org/abs/1703.10135

[2] L. Rabiner, "Speech synthesis by rule: An acoustic domain approach," *Bell System Technical Journal*, vol. 47, no. 1, pp. 17–37, 1968. [Online]. Available: http://dx.doi.org/10.1002/j.1538-7305.1968.tb00029.x

[3] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 373–376.

[4] S. King, "A beginners' guide to statistical parametric speech synthesis," *The Centre for Speech Technology Research University of Edinburgh*, June 2010.

[5] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[6] O. Watts, G. Henter, T. Merritt, Z. Wu, and S. King, *From HMMs to DNNs: Where Do the Improvements Come From?* IEEE, 3 2016, pp. 5505–5509.

[7] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastne, A. Courville, and Y. Bengio, "Char2Wav: End-to-End Speech Synthesis," https://openreview.net/forum?id=B1VWyySKx, February 2017.

[8] C. Weiss, L. C. Oliveira, S. Paulo, C. Mendes, L. Figueira, M. Vala, P. Sequeira, A. Paiva, T. Vogt, and E. André, "ecircus: building voices for autonomous speaking agents," in *Sixth ISCA Workshop on Speech Synthesis, Bonn, Germany, August 22-24, 2007*, P. Wagner, J. Abresch, S. Breuer, and W. Hess, Eds. ISCA, 2007, pp. 300–303. [Online]. Available: http://www.isca-speech.org/archive_open/ssw6/./ssw6_300.html

[9] P. Wagner, J. Abresch, S. Breuer, and W. Hess, Eds., *Sixth ISCA Workshop on Speech Synthesis, Bonn, Germany, August 22-24, 2007*. ISCA, 2007. [Online]. Available: http://www.isca-speech.org/archive_open/ssw6/

[10] Z. Wu, O. Watts, and S. King, *Merlin: An Open Source Neural Network Speech Synthesis System*, Sunnyvale, CA, USA, 9 2016, pp. 218–223.

[11] H. Kawahara, “Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds,” *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006.

[12] M. Morise, “D4c, a band-aperiodicity estimator for high-quality speech synthesis,” *Speech Communication*, vol. 84, pp. 57–65, 2016. [Online]. Available: https://doi.org/10.1016/j.specom.2016.09.001

[13] 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. E99-D, no. 7, pp. 1877–1884, 2016. [Online]. Available: http://search.ieice.org/bin/summary.php?id=e99-d_7_1877

[14] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, “Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based {F0} extraction: Possible role of a repetitive structure in sounds1,” *Speech Communication*, vol. 27, no. 3–4, pp. 187 – 207, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639398000855

[15] M. Morise, “Cheaptrick, a spectral envelope estimator for high-quality speech synthesis,” *Speech Communication*, vol. 67, pp. 1 – 7, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639314000697

[16] M. MORISE, “Error evaluation of an f0-adaptive spectral envelope estimator in robustness against the additive noise and f0 error,” *IEICE Transactions on Information and Systems*, vol. E98.D, no. 7, pp. 1405–1408. [Online]. Available: http://search.ieice.org/bin/summary.php?id=e98-d_7_1405

[17] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, “Conditional image generation with pixelcnn decoders,” *CoRR*, vol. abs/1606.05328, 2016. [Online]. Available: http://arxiv.org/abs/1606.05328

[18] J. Bradbury, S. Merity, C. Xiong, and R. Socher, “Quasi-recurrent neural networks,” *CoRR*, vol. abs/1611.01576, 2016. [Online]. Available: http://arxiv.org/abs/1611.01576

[19] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi, “Deep voice: Real-time neural text-to-speech,” *CoRR*, vol. abs/1702.07825, 2017. [Online]. Available: http://arxiv.org/abs/1702.07825

[20] F. P. Ribeiro, D. A. F. Florêncio, C. Zhang, and M. L. Seltzer, “CROWDMOS: an approach for crowdsourcing mean opinion score studies,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011,*

Prague Congress Center, Prague, Czech Republic, 2011, pp. 2416–2419. [Online]. Available: https://doi.org/10.1109/ICASSP.2011.5946971

[21] A. Graves, “Generating sequences with recurrent neural networks,” *CoRR*, vol. abs/1308.0850, 2013. [Online]. Available: http://arxiv.org/abs/1308.0850

[22] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, “Samplernn: An unconditional end-to-end neural audio generation model,” *International Conference on Learning Representations*, vol. abs/1612.07837, 2016. [Online]. Available: http://arxiv.org/abs/1612.07837

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

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *CoRR*, vol. abs/1603.04467, 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[25] Y.-Y. Chen, C.-H. Wu, and Y.-F. Huang, “Generation of emotion control vector using mds-based space transformation for expressive speech synthesis,” in *Interspeech 2016*, 2016, pp. 3176–3180. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-815

[26] M. Ghanim, “Multi-layer perceptron for multi-modal pain recognition,” 2015.

[27] H. Gunes, J. Vallverdú, and M. Pantic, “Automatic, dimensional and continuous emotion recognition,” *International journal of synthetic emotions*, vol. 1, no. 1, pp. 68–99, 1 2010, 10.4018/jse.2010101605.

[28] J. Andersson, S. Berlin, A. Costa, H. Berthelsen, H. Lindgren, N. Lindberg, J. Beskow, J. Edlund, and J. Gustafson, “Wikispeech - enabling open source text-to-speech for wikipedia,” in *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, 2016, pp. 93–99. [Online]. Available: https://doi.org/10.21437/SSW.2016-16

[29] Z. Malisz, H. Berthelsen, J. Beskow, and J. Gustafson, “Controlling prominence realisation in parametric dnn-based speech synthesis,” in *INTERSPEECH*, 2017.

[30] S. Ronanki, Z. Wu, O. Watts, and S. King, “A Demonstration of the Merlin Open Source Neural Network Speech Synthesis System,” United Kingdom, September 2016.

[31] K. Ito, “The lj speech dataset,” https://keithito.com/LJ-Speech-Dataset/, 2017.

[32] J. Ivens, *Vitor conhece o Pai Natal*. Edições ASA II.

[33] G. Delahaye, *Anita descobre a música*. Verbo Infantil.

[34] A. de Sant-Exupéry, *O Principezinho*. Relógi D'Água.

[35] B. Doumerc, *O Urso Simão*. Porto Editora.

[36] J. A. Lopes, G. Miguéis, J. L. Dias, A. Russo, A. F. Barata, F. Damião, and T. L. Fernandes, *Desenvolvimento de Competências Linguísticas em Jardim-de-Infância*. Edições Asa, 2006.