Controlling the Emotional Expressiveness of Synthetic Speech – a Deep Learning Approach

Noé Tits

noe.tits@umons.ac.be

December 18th, 2020

A dissertation submitted to the Faculty of Engineering
of the University of Mons, for the degree of Doctor of Philosophy in Engineering Science

Supervisor: Prof. T. Dutoit

This thesis was supported by the Fonds de la Recherche pour l'Industrie et l'Agriculture (F.R.I.A.), Belgium.

Jury members

Dr. **Nicolas d'Alessandro** - Hovertone
Dr. **Stéphane Dupont** - Université de Mons
Prof. **Thierry Dutoit** - Université de Mons, Supervisor
Dr. **Thomas Drugman** - Amazon
Prof. **Bernard Gosselin** - Université de Mons, President
Ir. **Emmanuel Jean** - Multitel
Prof. **Saïd Mahmoudi** - Université de Mons
Dr. **Vincent Pagel** - Acapela Group

“Il n’y a qu’une seule chose dont je suis sûr: c’est que je suis sûr d’une seule chose.”

Famille Tits

Acknowledgements

KEvin has been, from the start, a motivational motor and advisor for me throughout this project. He showed me the path to the art of writing a paper. He also tried to be my personal coach on other topics such as lifting wheights to build some muscle. Unfortunately, this part has not been that successful.

THierry, in addition to being an internationally renowned expert in speech processing, is also a person who is able to see in people what they are good at, allowing to put them in the right roles for them to be useful and feel useful. Thierry, despite his permanent stackoverflow state, always takes the time to think how to make things better. He has been a reference for me throughout this journey, both professionally and spiritually.

VIncent Pagel has been my greatest source of intuition and feedback through our discussions about the sometimes mysterious behaviour of the variety of models we have tried. Thank you for the time you spent discussing with me and telling me your best anecdotes.

IWould like to thank all the colleagues and friends of the lab with which I laughed a lot during dinner having the most absurd conversations it is possible to have. Thank you to ISIA's babies for bringing your enthusiasm

and jokes in the lab. A special thanks to those who listened to weird speech and laughter samples and/or read and reviewed this document for the better. Particularly Mathilde, you had the courage to read it entirely and gave me a lot of feedback regarding its structure and consistency.

I warmly thank my family and my friends for their caring and support. Thanks to all my friends for clearing my mind on all possible occasions. Mickaël, as a big brother, always has been and continues to be a model. Thanks to my parents for guiding and advising me and help make good decisions. I followed pretty closely his track. Our areas of interests seem to converge very well. And it seems that our little brother decided to share our passion for electrical engineering. A big thanks also to my father, Isa and André that provided me a lot of feedback for the redaction of this thesis.

Contents

List of acronyms

Part I

Introduction

Chapter 1

Scope and Contributions

Contents

This chapter is based on the following publication:

- Noé Tits. "A Methodology for Controlling the Emotional Expressiveness in Synthetic Speech - a Deep Learning approach". In: *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. 2019, pp. 1–5. DOI: 10.1109/ACIIW.2019.8925241

1.1 Introduction & Motivations

Text-To-Speech (TTS) synthesis systems, which synthesize speech from text, have been around for decades and have improved recently with the advent of new machine learning techniques such as Deep Neural Network (DNN). Companies such as Google and Amazon provide free DNN-based speech synthesis systems such as WaveNet[1] [71] or Amazon Polly.[2] These systems offer an excellent quality of speech obtained by analyzing tens of hours of neutral speech. However, the generated voices often fail to convey the emotional contents intended by the original speaker.

The objective of the present work is to produce *remarkable* voices, similar to those of professional actors, possessing a specific grain and a great capacity for expressiveness. This will make it possible to create virtual agents communicating in a more natural way, and thus to improve the efficiency of human-machine interaction.

To synthesize expressive speech, there is a need to *control* voice characteristics (aka features) to be able to modify intonation, rhythm, etc. The speech synthesis system has to give an access to *control parameters* that have an impact on voice characteristics of synthetic speech.

In the literature, there are three major approaches to synthesize speech [10]:

- a concatenation of pieces of recorded speech signal from a dataset;
- a modeling of how speech is mechanically produced;
- a statistical parametric model that is optimized by imitating the probability distributions of the dataset.

Depending on the synthesis technique, the generated voices will be more or less *natural* and the system may provide to the user more or less control on expressiveness in the produced speech. Speech *naturalness* is defined as speech output that sounds normal or natural to the listener [75, 84]. Synthesizers

[1] https://cloud.google.com/text-to-speech/
[2] https://aws.amazon.com/fr/polly/

based on concatenation can produce highly natural speech, but they provide little expressiveness control to the user, as the produced speech is strictly driven by the dataset used by the synthesizer. While modeling speech production offers control on many acoustic features, the resulting voice is unnatural. Indeed, modeling the phenomenon of speech production in a deterministic approach is done with many simplifying assumptions. Statistical approaches have been proven to allow a natural synthesis as well as a great flexibility, and therefore a potential to control many voice characteristics [123].

The most recent statistical approach uses DNN [122] and is the basis of new speech synthesis systems such as WaveNet [71] and Tacotron [111]. For that reason, in this work, the emphasis was placed on statistical parametric synthesis as the TTS system. There have been numerous studies to assess DNN-based systems ability to synthesize speech with high quality and naturalness such as Wavenet [71], Tacotron [111], Deep Convolutional Text-to-Speech (DCTTS) [93], WaveRNN [46], Char2Wav [91] and Deep Voice [3]. However, a major drawback of these approaches is the necessary large amount of speech data and high computational complexity. DCTTS needs less computational resources than the others. In [93], the authors explain that they were able to train a TTS model in 15 hours using a regular computer with two Graphical Processing Units (GPUs), resulting in nearly acceptable speech synthesis.

A closer look at the literature on producing speech of high quality and flexibility in terms of emotional content, however, reveals other difficulties. Access to a large amount of data remains a problem. The recording of high quality emotional speech datasets is expensive and time consuming. The amount of data available is therefore relatively limited compared to the needs of deep learning algorithms. This problem has previously been addressed using promising methods related to knowledge transfer such as transfer learning [74], fine-tuning [6] and multitask learning [78, 16].

Concerning the controllable aspect of TTS system, an important issue is the labelization of speech data with style or emotional information. Recent studies were conducted with unsupervised learning techniques to achieve controllable speech synthesis without the need of labels [1, 43, 40, 112, 90]. In [90], the

authors investigated such extension to the Tacotron speech synthesis system. The technique they propose is a system that learns an internal representation of information called latent space by encoding audio features into a vector that is concatenated with text information and then fed to Tacotron. These latent vectors model the remaining variability of speech after accounting for variation due to phonetics, speaker identity, and channel effects.

These works have provided evidence that building a latent space can lead to variables useful to control style in speech synthesis. In [43], the authors demonstrate that their system can synthesize spectrograms with different rhythms, speaking rates and fundamental frequencies from a single text, proving a control on these features.

A weakness of these studies is that they do not provide insights about the relationships between the resulting latent space and the audio features possible to control. We aim to fill this gap.

1.2 Contributions

As just discussed, to synthesize expressive speech, the main problem is the variability of the vocal expression of emotions. Deep Learning has been effective at handling complex data but requires a large amount of annotated data. The difficulty is then to annotate large databases with expressive metadata.

The goal of this work was to create an automatic system for annotating large expressive vocal databases, and synthesizing expressive speech. Our work plan, depicted in Figure 1.1 contains 4 main tasks:

- review and collect neutral and emotional speech data that is necessary for the other tasks;
- study the possibility to extract a representation of emotional expressiveness in speech with a Deep Learning architecture;
- build a system able to synthesize expressive speech based on the data collected, first based on specific styles, with an application to laughter;

- controlling an expressive speech synthesis system with a representation of emotional expressiveness, this information can come from labels or be extracted from a Deep Learning architecture.

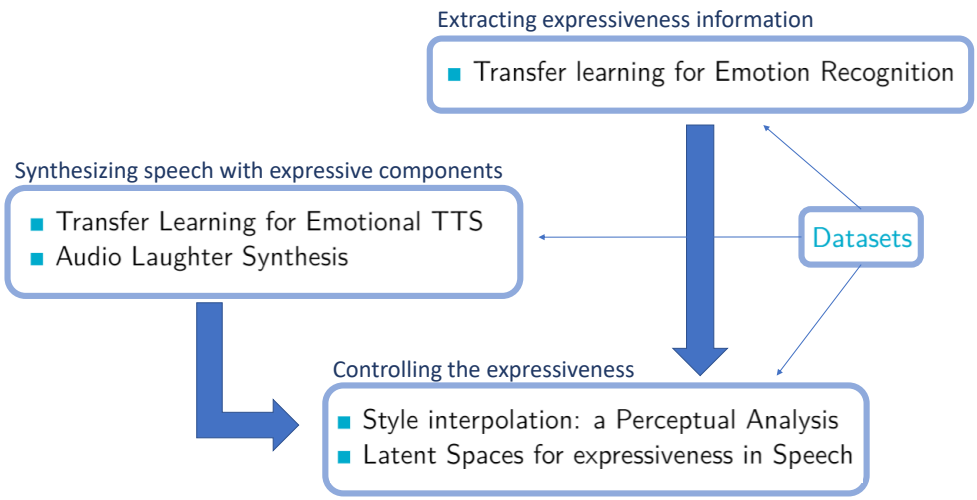

Figure 1.1. Plan of thesis contributions

This system was then integrated into a speech synthesis instrument from which it is possible to play an utterance as wished.

The original contributions of this thesis are as follows:

- To extract an emotion representation from speech, we present systems able to extract such a representation with supervised and unsupervised techniques. We also provide a way to visualize the trends of audio features in a latent space.
- Our contribution to answering the need for large quantity of data of Deep Learning algorithms is a review of existing datasets and the collection of EmoV-DB that will allow us to experiment on emotional TTS.
- For the synthesis itself, we study the impact of transfer learning from neutral TTS to emotional TTS and demonstrated the benefit of this technique.

- We propose an application of this transfer learning technique on Audio Laughter Synthesis.
- We give a brief analysis of the perception of speech by means of a simple method of continuous control of style in speech synthesis.
- We propose a methodology to build a system for speech synthesis that is able to control freely the expressiveness of speech via visualization and interpretation of a learned latent space. We evaluate its performance with two kinds of styled speech datasets: the first one has classes of style different from each other, the second has more continuous variations of diverse expressive speech.
- We present an integration of the system in an interface to have a user-friendly tool to generate speech with an access to the different controls (text and expressiveness representation).
- We propose a perceptual experiment for assessing the controllability of a Controllable Expressive TTS model with an adaptation of the interface. This experiment studies if a user is able to find a sample with an expressiveness similar to a reference inside the 2D interface.

1.3 Organization of the Dissertation

- Chapter 2 presents the theoretical notions that the reader should understand about expressive speech processing and associated paradigms in digital signal processing, machine learning systems and psychological models of emotion expressiveness.
- Chapter 3 describes existing expressive datasets and their different characteristics and present EmoV-DB, a dataset recorded in collaboration with Northeastern University containing emotional speech of several categories and several speakers.
- In Chapter 4, an analysis of emotional content in speech using a Deep Learning based Automatic Speech Recognition system as feature extractor is presented.
- Chapter 5 presents a study on the efficiency of domain adaptation from neutral speech synthesis to emotional speech synthesis.

- Chapter 6 is an application of the method proposed in Chapter 5 to Audio Laughter synthesis, i.e., a domain adaptation from speech synthesis to laughter synthesis.
- Chapter 7 describes an analysis of style interpolation in the context of styled speech synthesis with specific categories. In this study, we evaluate the ability of a Deep Learning (DL) TTS system to interpolate between neutral speech and styled speech without having labels of intermediate styles.
- In Chapter 8, we compare different latent spaces aiming at representing vocal expressiveness of a dataset containing different categories of styles. An acoustic analysis is performed to study the relationship between the designed latent spaces and style categories of the voice. An extended analysis of one of the systems is then performed in the context of continuous variations of expressiveness from a speech dataset containing recordings of various and expressive audiobooks. The synthesis abilities of the system are also assessed.
- Finally, Chapter 9 shows a proof of concept of a controllable expressive speech synthesis system based on the research results described in previous chapters, and presents a perceptual experiment assessing the controllability of such a system. More specifically, this experiment studies if a user is able to find a sample with an expressiveness similar to a reference inside the 2D interface.

Chapter 2

Theoretical background

Contents

This chapter is based on the following publication:

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "The Theory behind Controllable Expressive Speech Synthesis: A Cross-Disciplinary Approach". In: *Human-Computer Interaction*. IntechOpen, 2019. DOI: 10.5772/intechopen.89849. URL: http://dx.doi.org/10.5772/intechopen.89849

As part of the Human-Computer Interaction field, Expressive speech synthesis is a very rich domain as it requires knowledge in areas such as machine learning, signal processing, sociology, psychology.

In this chapter, we will focus mostly on the technical side. From the recording of expressive speech to its modeling, the reader will have an overview of the main paradigms used in this field, through some of the most prominent systems and methods.

We explain how speech can be represented and encoded with audio features. We present a history of the main methods of Text-to-Speech synthesis: Concatenative, Speech Production Modeling and Statistical Parametric Speech Synthesis. Finally, we focus on the last one, with the last techniques modeling Text-to-Speech synthesis as a sequence-to-sequence problem. This enables the use of Deep Learning blocks such as Convolutional and Recurrent Neural Networks as well as Attention Mechanism. The last part of the chapter intends to assemble the different aspects of the theory and summarize the concepts.

2.1 Introduction

Controllable Expressive Speech Synthesis is the task of generating expressive speech from a text with control on prosodic features. This task is positioned in the emerging field of Affective Computing and more particularly at the intersection of three disciplines:

- Expressive speech analysis (Section 2.2), which provides mathematical tools to extract useful characteristics from speech depending on the task to be performed. Speech is seen as a signal, such as images, text, videos or any kind of information coming from any source. As such, it can be characterized by a time series of features.
- Expressive speech modeling (Section 2.3), modeling human emotions and their impact on the speech signal. Speech is considered here as a means of communication between humans.

- Expressive speech synthesis (Section 2.4), which consists in generating expressive speech from text and for which machine learning tools have become ubiquitous, especially Hidden Markov Models (HMMs) and more recently DNNs. The field of Machine Learning allows machines to learn how to solve a given task. This field borrows from an ensemble of statistical models used for representation or transformation data. It also uses concepts from Information Theory to measure distances between probability distributions.

2.2 Expressive Speech Analysis

2.2.1 Digital Signal Processing

A signal is a variation of a physical quantity carrying information. The acoustic speech signal is converted into an electrical signal by a microphone. An acoustic signal is a variation of pressure in a fluid that the human perceives through the sense of hearing. This signal is mono-dimensional because it can be represented by a scalar function that expresses pressure depending time.

The electrical signal generated by the microphone is an analog signal. In order to process it with a digital machine, it must be digitized. This is done by electronic systems called analog-to-digital converters that discretize time and quantize the pressure signal to convert it into a digital signal. After some processing of the digitized signal, a digital-to-analog converter can be used to convert the processed digital signal back into an analog signal. This analog electrical signal can then be converted into an acoustic signal though loudspeakers or earphones to make it available to human ears. These steps are represented in Figure 2.1.

Digital signal processing [21] is the set of theories and techniques for analyzing, synthesizing, quantifying, classifying, predicting, or recognizing signals, using digital systems.

A digital system receives as input a sequence of quantized samples $\{x(0), x(1), x(2), ...\}$, noted as $\{x(n)\}$, and produces as output a sequence of samples $\{x(n)\}$ after

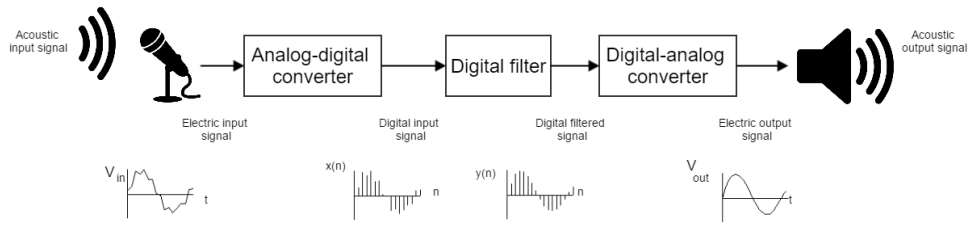

Figure 2.1. Digital signal processing for acoustic signals. The acoustic signal is converted into an electric signal by means of a microphone. This signal is then sampled and quantized to produce a digital signal. This digital signal can then be processed by a computer. In the end, to listen to the resulting signal, it has to be converted back to an analog signal and played via a loudspeaker.

application of a series of algebraic operations.

A digital filter is a linear and invariant digital system. Let us consider a digital system that receives the sample sequences $\{x_1(n)\}$ or $\{x_2(n)\}$ as input. This system will produce the sample sequences $\{y_1(n)\}$ for the input $\{x_1(n)\}$ and $\{y_2(n)\}$ for the input $\{x_2(n)\}$. This system is linear if it produces the output $\{\alpha y_1(n) + \beta y_2(n)\}$ when it receives the sequence $\{\alpha x_1(n) + \beta x_2(n)\}$ as input. A digital system is said to be invariant if shifting the input sequence by n_0 samples also shifts the output sequence by n_0 samples.

These linear and invariant digital systems can be described by equations of the type:

$$\begin{aligned} y(n) + a_1 y(n-1) + a_2 y(n-2) + ... + a_N y(n-N) = \\ b_0 x(n) + b_1 x(n-1) + ... + b_M x(n-M) \end{aligned} \tag{2.1}$$

or

$$y(n) + \sum_{i=1}^{N} a_i y(n-i) = \sum_{i=0}^{M} b_i x(n-i) \tag{2.2}$$

Put differently, the output $y(n)$ is a linear combination of the last N outputs, the input $x(n)$, and the M previous inputs. A digital filter is therefore determined if the coefficients a_i and b_i are known. A filter is called non-recursive if only the inputs are used to compute $y(n)$. If at least one of the previous output samples is used, it is called a recursive filter.

2.2.2 Speech Features

Speech is a signal carrying a lot of information. These expand from the sequence of words used to create a sentence, to the tone of voice used to utter this sentence. Depending on the task, it is often needed to select the useful part of the information and discard the rest. Also, the speech can carry noise before reception. That's why an important step in speech analysis is to extract descriptors or features that are relevant to the task of interest.

There exist many different feature spaces that describe speech information. In this section, we give an intuitive explanation of the ones widely used in Deep Learning architectures.

Power spectral density and spectrogram

Fourier analysis demonstrates that any physical signal can be decomposed into a sum of sinusoids of different frequencies. The power spectral density of a signal describes the amount of power carried by the different frequency bands of this signal. This range of frequencies may be a discrete value set or a continuous frequency spectrum. In the field of digital signal processing, this power spectral density can be calculated by the Fast Fourier Transform (FFT) algorithm.

The graph of the power spectral density allows the visualization of the frequency characteristics of a signal such as the fundamental frequency of a periodic signal and its harmonics. A periodic signal is a repetition of a sequence of samples of a given length called period, noted T_0. The number of periods per unit of time that repeats is the fundamental frequency, i.e., $F_0 = \frac{1}{T_0}$. Harmonics are the multiple frequencies of the fundamental. These frequencies have an important power density and present therefore extrema in the power spectral density.

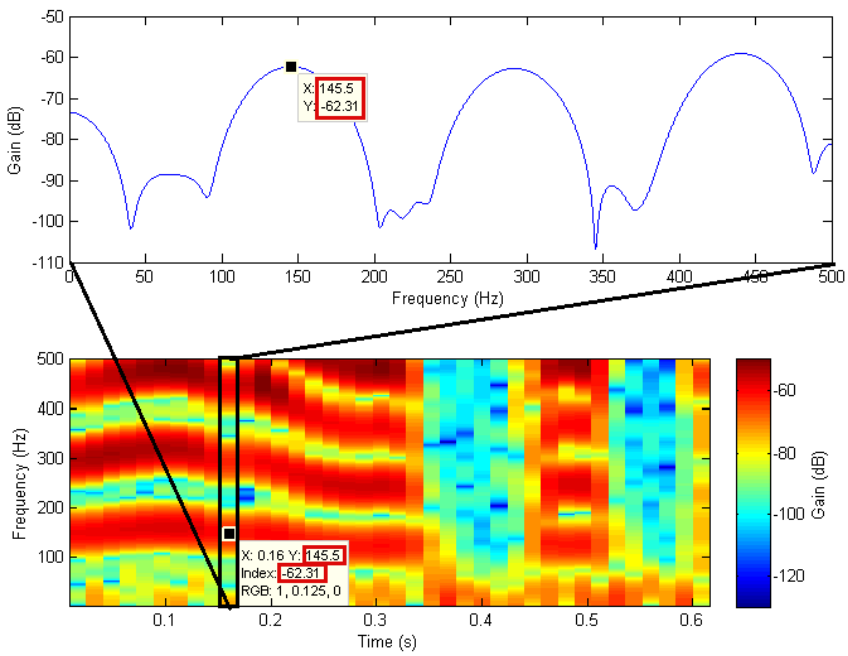

Figure 2.2. Spectrum (top) and spectrogram (bottom) of a speech segment. The spectrum is the Fourier transform of a frame of a signal that in this case comes from speech. It represents the magnitude of sine waves that compose the signal. The spectrogram is obtained by concatenating spectrums of frames across time in a matrix. It is represented as a heat map in which the color corresponds to the magnitude.

An example of power spectral density is shown in the upper part of Figure 2.2. The first maximum is at the fundamental frequency which is 145.5 Hz. The other maxima are the harmonics.

When the signal's characteristics are evolving over time, as with the voice signal, the spectrogram can be used to visualize this evolution. The spectrogram represents the power spectral density over the time. An example of power spectrogram is shown in the lower part of Figure 2.2. The x-axis is time and the y-axis is frequency. The colors correspond to the power density. A color scale is given on the right of the graph. The spectrogram is thus constructed by juxtaposing power spectral density functions computed on every frame as suggested in Figure 2.2.

Mel-Spectrogram

The Mel-Spectrogram is a reduced version of the spectrogram. The use of this feature is very widespread for machine learning-based systems in general and for Deep learning-based TTS in particular.

The intuition behind this feature is to compress the representation of the speech in the higher values of the frequency domain based on the fact that the human ear is sensitive to some frequencies more than others. The mel scale is an experimental function representing the sensitivity of the human ear depending on the frequency.

The conversion of frequency f in mel-frequency m is:

$$m = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.3}$$

Figure 2.3 shows the curve of the mel scale as a function of the frequency. As one can observe, an interval of low frequencies is mapped to a larger interval of mel values than for high frequencies. As an example, the interval $[0, 2000]$ Hz is mapped to more than 1500 mel while the interval $[8000, 10000]$ Hz is mapped to less than 300 mel.

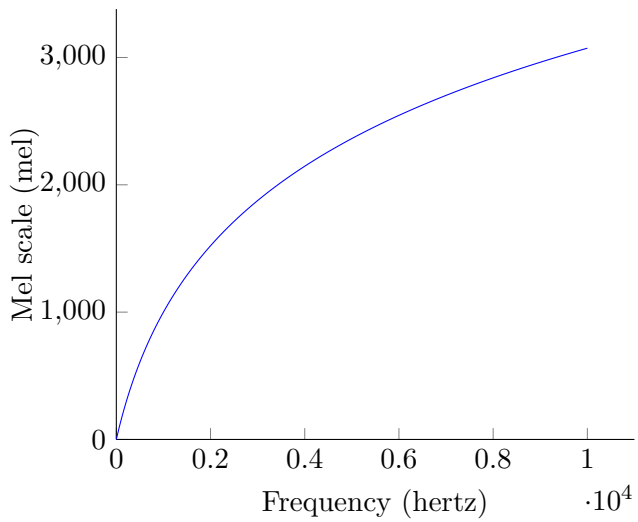

Figure 2.3. Mel scale representing the perception of frequencies of an acoustic signal by a human ear.

2.3 Modeling of Emotion Expressiveness

Emotion modeling is one of the main challenges in developing more natural human-machine interfaces. Among the many previously proposed approaches, those based on Ekman's model [22] or on Russels' model [86] are most widely used in applications. A first representation is Ekman's six basic emotions model [22] which identifies anger, disgust, fear, happiness, sadness and surprise as six basic categories of emotions from which the other emotions may be derived.

Emotions can also be represented in a multidimensional continuous space like in the Russels circumplex model [86] (see Figure 2.4[1]) . This model makes it possible to better reflect the complexity and the variations in the expressions, unlike the category system. The two most commonly used dimensions in the literature are arousal, corresponding to the level of excitation and valence, cor-

[1] https://commons.wikimedia.org/wiki/File:Valence-Arousal_Circumplex.jpg - imagine-it.org/CC-BY-3.0

responding to the pleasantness level or positiveness of the emotion. A third dimension is sometimes added: dominance corresponding to the level of power of the speaker relative to the listener.

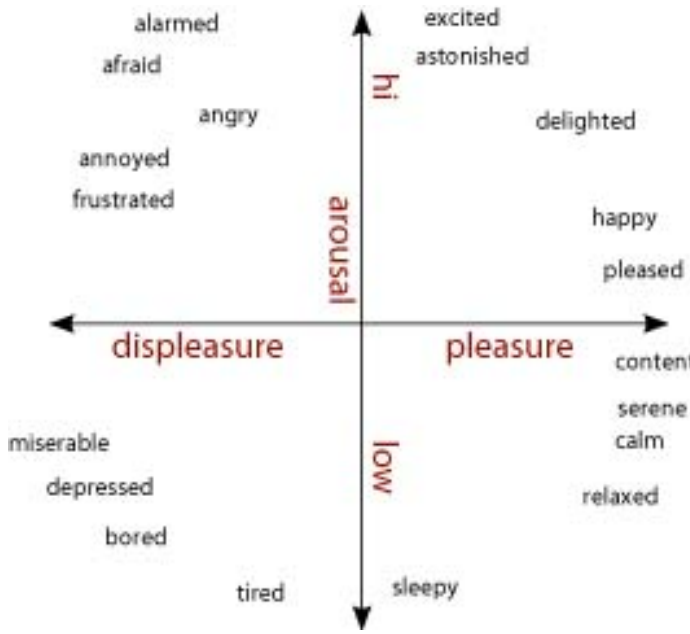

Figure 2.4. Russel's Circumplex of Affect. A psychological model of how emotions are organised. This model assumes that emotions can be localized in a space of two dimensions named Valence corresponding to a degree of pleasantness and Arousal corresponding to a degree of activation.

Another way of representing emotions is based on ranking with a relative preference method to annotate emotions rather than labeling them with absolute values [117]. The argument for this method is that humans are not reliable for assigning absolute values to subjective concepts. However they are better at discriminating between elements shown to them. Therefore, the design of perception tasks, e.g., about emotion or style in speech, should take this into account by asking participants to solve comparison tasks rather than rating tasks.

It is important to note that many other approaches exist [5] and it is a difficult question to know what approach should be used in applications in the field of Human-Computer Interaction. Indeed these psychological models of affect are propositions of explanations of how emotions are expressed. But these propositions are difficult to assess in practice.

Humans express their emotions via various channels: face, gesture, speech, etc. Different people will express and perceive emotions differently depending on their personality, their culture and many other aspects. For developing applications, one has therefore to take assumptions to reduce its scope and choose one approach of emotion modeling.

In this chapter we are interested in how the expressive speech synthesized will be perceived. It is therefore reasonable to begin by choosing a language and assuming the origin of the synthesized voice.

Research has recently evolved into systems using, without preprocessing, the signal or spectrogram of the signal as input: the neural network learns the features that best correspond to the task it is supposed to perform on its own. This principle has been successfully applied to the modeling of emotions, currently constituting the state of the art in speech emotion recognition [63, 105].

2.4 Expressive Speech Synthesis

2.4.1 A brief History of Speech Synthesis Techniques and How to Control Expressiveness

The goal behind a speech synthesis system is to generate an audio speech signal corresponding to any input text. A sentence is constituted of characters and a human knows how these characters should be pronounced. Indeed, a human knowing the rules of the language knows what sequence of *phonemes* (aka *phones*) he has to utter to read a sentence aloud. If we want a machine to be

able to generate speech signal from text, we have to teach it, or program it to do the same.

Such systems have been developed for decades and many different approaches have been used. Here we summarize them in three categories: Concatenation, Source-Filter modeling and Statistical Parametric Speech Synthesis.[2]

Concatenation

This approach is based on the concatenation of pieces of audio signals corresponding to different phonemes. This method is segmented in several steps. First, the characters should be converted in the corresponding phonemes to be pronounced. A simplistic approach is to assume that one letter corresponds to one phoneme for example [31]. Then the computer must know what signal corresponds to a phoneme. A possibility to solve this problem is to record a database containing all the existing phonemes in a given language.

However concatenating phones one after another leads to very unnatural transitions between them. In the literature, this problem was tackled by recording successions of two phonemes, called diphones, instead of phones [21]. All combinations of diphones are recorded in a dataset. The generation of speech is then performed by concatenation of these diphones.

In this approach, many assumptions are not met in practice. First, a text processing has to be performed, indeed, text consists of punctuation, numbers, abbreviations, etc. Moreover, the letter to sound relationship, assuming that one letter corresponds to one sound, is not respected in English and in many other languages. The pronunciation of words often depends on the context. Also, concatenating phones leads to a chopped signal and prosody of the generated signal is unnatural.

To have a control on expressiveness with diphone concatenation techniques, it is possible to change Fundamental Frequency (F_0) and duration with signal

[2] However the state of the art is more diverse and complex. It contains many variants and hybrid approaches between them.

processing techniques implying some distortion on the signal. It is difficult to control other speech features without altering the signal leading to unnatural speech.

Another approach that is also based on the concatenation of pieces of signal is *Unit Selection*. Instead of concatenating phones (or diphones), larger parts of words are concatenated. An algorithm has to select the best units according to criteria: few discontinuities in the generated speech signal, a consistent prosody, etc.

For this purpose, a much larger dataset must be recorded containing a large variety of different combinations of phone series. The machine must know what part of signal corresponds to what phoneme, which means it has to be annotated by hand accurately. This annotation process is time consuming. Today there exist tools to do this task automatically. But this automation can in fact be done at the same time as synthesis as we will see later.

The advantages of this method is that the signal is less altered and most of the transitions between phones are natural because they are coming as is from the dataset.

With this method, a possibility to synthesize emotional speech is to record a dataset with separate categories of emotion. In synthesis, only units coming from a category will be used [87]. The drawback is that it is limited to discrete categories without any continuous control.

Speech production modeling

Anatomically, the speech signal is generated by an excitation signal generated in the larynx. This excitation signal is transformed by resonance through the vocal tract (guttural, oral and nasal cavities) which acts as a filter. If this excitation signal is generated by glottal pulses, then a voiced sound is obtained. Glottal pulses are generated by a series of openings and closures of vocal cords or vocal folds. The vibration of the vocal cords has a fundamental frequency. As opposed to voiced sounds, when the excitation signal is a simple flow of

exhaled air, it is an unvoiced sound.

The source-filter model is a way to represent speech production which uses the idea of separating the excitation and the resonance phenomenon in the vocal tract. It assumes that these two phenomena are completely decoupled. The source corresponds to the glottal excitation and the filter corresponds to the vocal tract. This principle is illustrated in Figure 2.5[3].

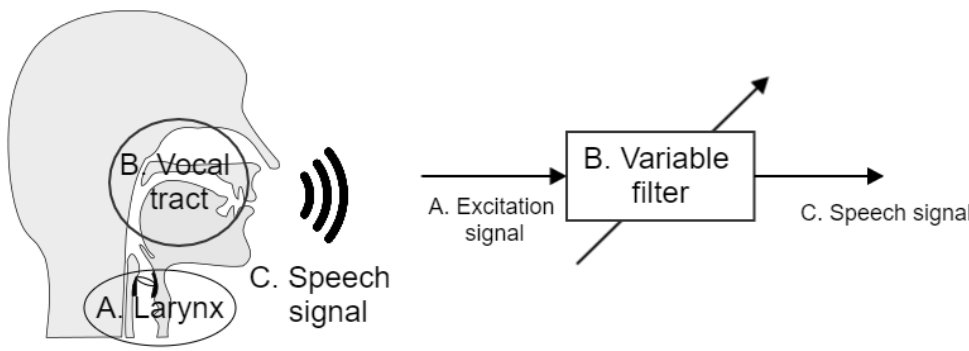

Figure 2.5. Diagram describing voice production mechanism and source-filter model. The larynx is the source of vibrations in the air and the vocal tract acts like a filter varying along time to give the speech signal.

An example of Source-Filter modeling is the linear prediction model. The Linear Prediction (LP) model [39] uses this Source-Filter theory assuming that the speech is the output signal of a recursive digital filter, when an excitation is received at the input. In other words, it is assumed that each sample can be predicted by a linear combination of the last p samples. The linear predictive coding works by estimating the coefficients of this digital filter representing the vocal tract. The number of coefficients to represent the vocal tract has to be chosen. The more coefficients we take, the better the vocal tract is represented, but the more complex the analysis will be. The excitation signal can then be computed by applying the inverse filter on the speech signal.

[3]Vocal tract image from: https://en.wikipedia.org/wiki/User:Tavin#/media/File:VocalTract.svg - Tavin/CC-BY-3.0

In synthesis, this excitation signal is modeled by a train of pulses for voiced sounds and white noise for unvoiced sounds. In reality, the mechanics of the vocal folds are more complex, making this assumption too simplistic. There exist other source-filter models that take less simplistic assumptions, with a more complex excitation signal mixing deterministic and stochastic components [20, 32].

The vocal tract is a variable filter. Depending on the shape we give to this vocal tract, we are able to produce different sounds. A filter is considered constant for a short period of time and a different filter has to be computed for each period of time.

This approach has been successful to synthesize intelligible speech but not natural human sounding speech. For expressive speech synthesis, this technique has the advantage of giving access to many control parameters of speech.

The approach used in [12] to discover how to control a set of acoustic features to obtain a desired emotion was done through perception tests. A set of sentences were synthesized with different values of these features. These sentences were then used in listening tests in which participants were asked to answer questions about the emotion they perceived. Based on these results, values of the different features were associated with the emotion expressions.

Statistical Parametric Speech Synthesis

Statistical Parametric Speech Synthesis (SPSS) is less based on knowledge, and more based on data. We take less simplistic assumptions on the speech generation and rely more on the statistics of data to explain how to generate speech from text.

The idea is to teach a machine the probability distributions of signal values depending on the text that is given. We generally assume that generating the values that are most likely is a good choice. We thus use the Maximum Likelihood principle (see Section 2.4.4).

These probability distributions are estimated based on a speech dataset. To be a good estimation of the reality, this dataset must be large enough. The first successful SPSS systems were based on HMMs and Gaussian Mixture Models (GMMs). The most recent statistical approach uses DNN [122] which is the basis of new speech synthesis systems such as WaveNet [71] and Tacotron [111]. The improvement provided by this technique [114] comes from the replacement of decision trees by DNNs and the replacement of state prediction (HMM) by frame prediction.

In the rest of this chapter, we focus on this approach of Speech Synthesis. Section 2.4.2 details the different modules of the typical pipeline of SPSS. Section 2.4.3 explains Deep Learning focusing on Speech Synthesis application and Section 2.4.4 presents concepts of information theory and probabilities important for speech processing.

2.4.2 Typical pipeline of Statistical Parametric Speech Synthesis

In this section, we detail the main building blocks of SPSS approach briefly described in Section 2.4.1. As depicted in Figure 2.6, this pipeline can be segmented in three main parts: a text processing front-end, an acoustic model and a vocoder.

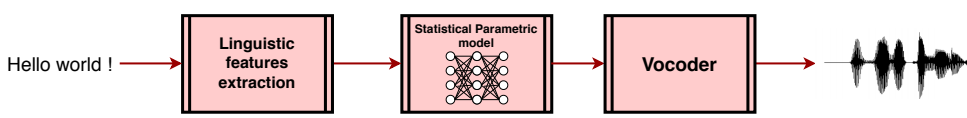

Figure 2.6. Pipeline of a typical SPSS composed of three blocks. The first block receives text in input and rules of the language are applied to extract linguistic information such as phonemes. The second block is the acoustic model, it predicts acoustic features from linguistic information. The last block generates the waveform from acoustic features.

Text processing The role of this block is to extract linguistic features from text. This block typically has to apply the rules of the language to extract the right phonemes depending on the context, converting numbers to letters, handle punctuation and many details and exceptions that exist.

In recent TTS systems using DL architectures, raw characters are often used as linguistic features as these statistical models showed to be able to handle more complex relationships.

Acoustic Modeling This part is responsible for mapping text information to acoustic features. The problem is formulated as a regression problem in which we wish to find a function able to map inputs to outputs. The inputs are linguistic features extracted by the first block, and the outputs are acoustic features from which it is possible to reconstruct speech waveform with a vocoder.

Vocoder The vocoder is typically a signal processing module that synthesizes the waveform signal from a set of acoustic features. Many vocoders actually use the source-filter theory described in Section 2.4.1. A standard vocoder is WORLD [68], that decomposes the speech waveform in F_0, spectral envelope and aperiodicity information. A vocoder is capable of waveform reconstruction from these features.

Another approach is to use magnitude spectrogram as an acoustic feature set and use a phase estimation algorithm, e.g., Griffin-Lim phase estimation algorithm [38] and reconstruct the waveform signal with inverse Fourier Transform.

Finally, last approaches of the literature rely on DNN to replace this signal processing block by a statistical model that predicts waveform samples from speech features, e.g., Wavenet [71], WaveRNN [46] or MelGAN [54].

2.4.3 Deep Learning for Speech Synthesis

Machine Learning consists of teaching a machine to perform a specific task, using data. In this chapter, the task we are interested in is Controllable Expressive Speech Synthesis. The mathematical tools for this come from the field of Statistical Modeling.

Deep Learning is based on the optimization of a mathematical model which is a parametric function. This model is optimized or *trained* by comparing its predictions to ground truth examples taken from a dataset. This comparison

is based on a measure of similarity or error between a prediction and the true example of the dataset. The goal is then to minimize the error or maximize the similarity. This can always be formulated as the minimization of a loss function.

To find a good loss function, it is necessary to understand the statistics of the data we want to predict and how to compare them. For this, concepts from information theory are used.

Different operations and Architectures

The form of the mathematical function used to process the signal can be constituted of lots of different operations. Some of these operations were found very performant in different fields and are widely used. In this section, we describe some operations relevant for speech synthesis. In Deep Learning, the ensemble of the operations applied to a signal to have a prediction is called *Architecture*. There is an important research interest in designing architectures for different tasks and data to process. This research reports empirical results comparing the performance of different combinations. The progress of this field is directly related to the computation power available on the market.

Historically, the root of Deep Learning is a model called Neural Network. This model was inspired by the role of neurons in the brain that communicate with electrical impulses and process information. Since that, more recent models drove away from this analogy and evolved depending on their actual performance.

Fully connected neural networks Fully connected neural networks are successions of linear projections followed by non-linearities (sigmoid, hyperbolic tangent, etc.) called layers.

$$h = f_h(W_h x + b_h) \tag{2.4}$$

x: input vector

h: hidden layer vector

W_h and b_h: parameter matrices and vector of layer h

f_h: Activation function of layer h

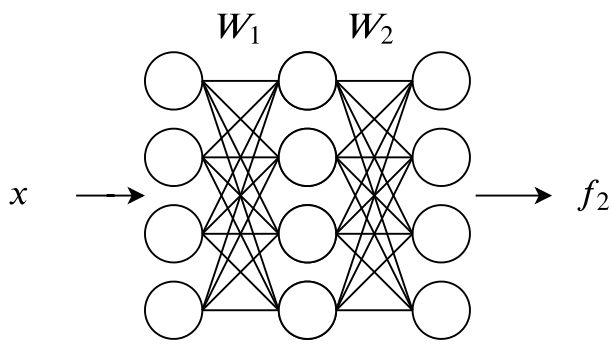

Figure 2.7. Diagram of fully connected network equation.

More layers implies more model parameters and thus a more complex model. It also means more intermediate representations and transformation steps. It was shown that deeper Neural Networks (more layers) performed better than shallow ones (fewer layers). This observation led to the names DNNs and Deep Learning. A complex model is capable of modeling a complex task but is also more costly to optimize in terms of computation power and data.

Convolutional neural network Convolutional Neural Network (CNN) [58, 36] refers to the operation of convolution and reminds the convolution filters of signal processing (see Equation 2.5). A convolution layer can thus be seen as a convolutional filter for which the coefficients were obtained by training the Deep Learning architecture.

$$g(x,y) = \omega * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x-s, y-t) \tag{2.5}$$

f: input matrix

g: output matrix

ω: convolutional filter weights

Convolutional filters were studied in the field of image processing. We know what filters to apply to detect edges, to blur an image, etc. In practice, the operation implemented is often a correlation which is the same operation except that the filter is not flipped. Given that the model filter parameters are optimized during training, the flipping part is useless. We can just consider that the filter optimized with a correlation implementation is just the flipped version of the one that would have been computed if convolution was implemented.

For speech synthesis, convolutional layers have been used to extract a representation of linguistic features and predict spectral speech features. For temporal signals such as speech, one dimensional convolution along the time axis models well time dependencies. As layers are stacked, the receptive field increases proportionally. In speech, there are long-term dependencies in the signal, e.g., in the intonation and emphasis of some words. To model these long-term dependencies, dilated convolution [71] was proposed. It allows an increase of the receptive field exponentially instead of proportionally with the number of layers.

Recurrent Neural Network Recurrent Neural Network (RNN) [36] involves a recursive behaviour, i.e., having an information feedback from the output to the input. The principle is illustrated in Figure 2.8 and is analogous to recursive filters. Recursive filters are filters designed for temporal signals because they are able to model causal dependencies. It means that at a given time t, the value depends on the past values of the signal.

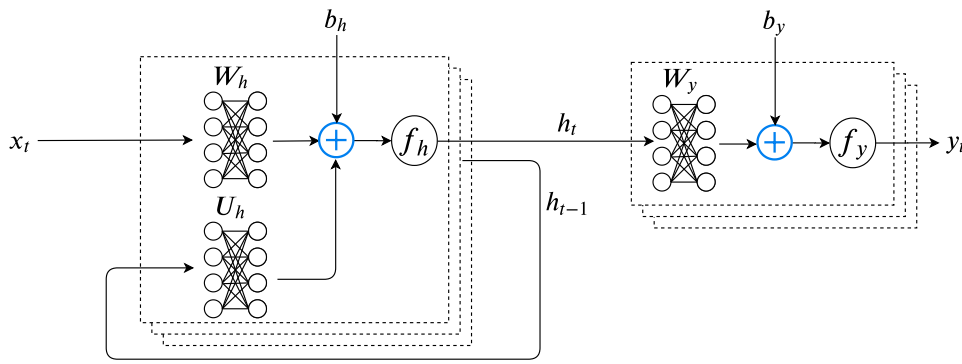

Figure 2.8. Diagram of RNN equations.

$$h_t = f_h(W_h x_t + U_h h_{t-1} + b_h) \tag{2.6}$$

$$y_t = f_y(W_y h_t + b_y) \tag{2.7}$$

x_t: input vector

h_t: hidden layer vector

y_t: output vector

W_h, U_h and b_h: parameter matrices and vector of layer h

f_h and f_y: Activation function of layer h and output layer, respectively

Encoder and Decoder An encoder is a part of a neural network that outputs a hidden representation (or latent representation) from an input. A decoder is a part of a neural network that retrieves an output from a latent representation.

When the input and the output is the same, we talk about auto-encoders. The task in itself is useless, but the interesting part here is the latent representation. The latent space of an auto-encoder can provide interesting properties such as a lower dimensionality, meaning a compressed representation of the initial data or meaningful distances between examples.

Sequence-to-sequence modeling and Attention Mechanism A sequence-to-sequence task is about converting sequential data from one domain to another, e.g., from a language to another (translation), from speech to text (speech recognition) or from text to speech (speech synthesis).

First Deep Learning architectures for solving sequence-to-sequence tasks were based on encoder-decoder with RNNs called RNN transducers. Other techniques were found to outperform this. The use of Attention Mechanism was found beneficial [81]. Attention Mechanism was first developed in the field of computer vision. It was then successfully applied to ASR and then to TTS.

In the Deep Learning architecture, a matrix is computed and used as weighting on the hidden representation at a given layer. The weighted representation is fed to the rest of the architecture until the end. This means that the matrix is asked to emphasize the part of the signal that is important to reduce the loss. This matrix is called the Attention matrix because it represents the importance of the different regions of the data.

In computer vision, a good illustration of this mechanism is that for a task of classification of objects, the attention matrix has high weights for the region corresponding to the object and low weights corresponding to the background of the image.

In ASR, this mechanism has been used in a so-called Listen, Attend, and Spell (LAS) [17] setup. An important difference compared to the previous case is the sequential nature of the problem. There must be an information feedback to have a recursive kind of architecture and each time step must be computed based on previous time steps.

LAS designates three parts of the Deep Learning architecture. The first one encodes audio features in a hidden representation. The role of the last one is to generate text information from a hidden representation. Between this encoder and decoder, at each time step, an Attention Mechanism computes a vector that will weight the text encoding vector. This weighting vector should give importance to the part of the utterance that the architecture should pay attention to in order to generate the corresponding part of speech.

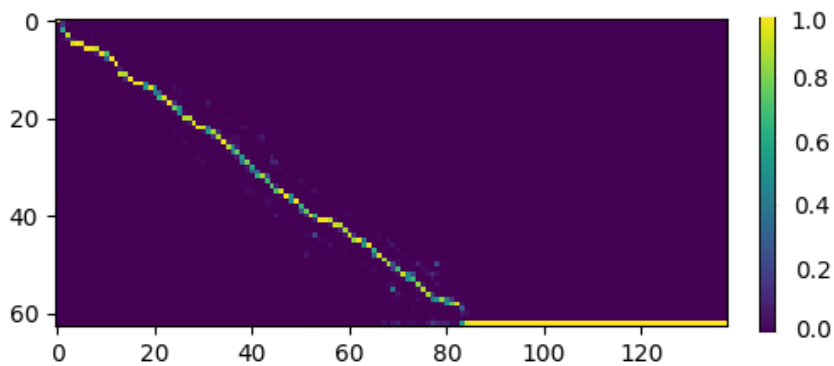

Figure 2.9. Alignment plot. The y-axis represents the text character indices and the x-axis represents the audio frame feature indices. The color scale corresponds to the weight given to a given character to predict a given audio frame. The path shows which character is important to focus on at each time step of the audio features.

An Attention plot (see Figure 2.9) of a generated sentence can be constructed by juxtaposing all the weighting vectors computed during the generation of a sentence. The resulting matrix can then be represented by mapping a color scale on the values contained.

This attention plot shows an attention path, i.e., the importance given to text characters along the audio output timeline. As it can be observed in Figure 2.9, this attention path should have a close to diagonal shape. Indeed the two sequences have a close chronological relationship.

Transfer Learning, Domain Adaptation and Fine-Tuning As previously explained, Deep Learning allows a machine to perform a specific task, using data. For this purpose, a model is trained by comparing its predictions to ground truth examples taken from a dataset. A major drawback of DL is the need for large quantities of data, expensive and/or difficult to collect, and the computational complexity of training models from scratch.

A useful paradigm that tackles this drawback is that of transfer learning or domain adaptation [74, 118, 95], where the knowledge learned for one task is exploited to improve performance on another task.

Transfer Learning can be achieved by a broad varieties of techniques to improve the performance of Machine Learning algorithms in general [74, 115], but more specifically for Deep Learning, due to this need of data. In [95], they identify different families of deep transfer learning techniques.

Among all these techniques, a technique for transfer learning, called *fine-tuning*, consists of continuing the training process of the parameters of a model pre-trained on a related task. The number of parameter layers that are fine-tuned can vary depending on the extent to which the task are related.

There are many other ways to do transfer learning. In the field of speech synthesis, an example is [45] in which they train a system on the task of speaker identification and use it to extract a representation of speaker voice characteristics in multi-speaker speech synthesis system.

Evolution of State-of-the-art Deep Learning-based TTS

Merlin toolkit There are more and more DL-based TTS systems developed. Merlin [116] toolkit has been an important tool to investigate the use of DNNs for speech synthesis. The first models developed within Merlin were based only on Fully connected neural networks. One DNN was used to predict acoustic features and another one to predict phone durations. It was a first successful

attempt that outperformed other statistical approaches at the time.

In [114], authors study the components of these DNN approaches that are responsible of the improvement over HMM-based TTS. They conclude that the switch from decision-tree to DNN regression, and the change from state- to frame-level targets for the regression are responsible for the greatest perceptual gains.

However a major limit of fully-connected DNNs is that they are not designed to model timed data, and therefore ignore the auto-regressive nature of speech signals. More recent architectures, including CNN and RNN such as Long Short Term Memory (LSTM) have therefore been proposed. Merlin toolkit allowed to study early stage sequence-to-sequence models with an encoder-decoder setup.

Waveform synthesis models A different approach to speech synthesis has emerged a few years ago. It consists of generating directly the audio waveform from linguistic features with a DL model using the auto-regressive nature of audio signals. From the pipeline presented in Figure 2.6, this model takes the role of acoustic model and vocoder at the same time.

Wavenet [71] is a stack of Gated Convolutional Units (GCUs). As stated in [71] its equation is:

$$\vec{z} = \tanh\left(W_{f,k} * \vec{x}\right) \odot \sigma\left(W_{g,k} * \vec{x}\right), \tag{2.8}$$

where $*$ denotes a convolution operator, $\odot$ denotes an element-wise multiplication operator, $\sigma(\cdot)$ is a sigmoid function, k is the layer index, f and g denote filter and gate, respectively, and W is a learnable convolution filter.

WaveRNN [46] is another autoregressive model based on a single-layer RNN that matches the performance of Wavenet while being more computationally efficient.

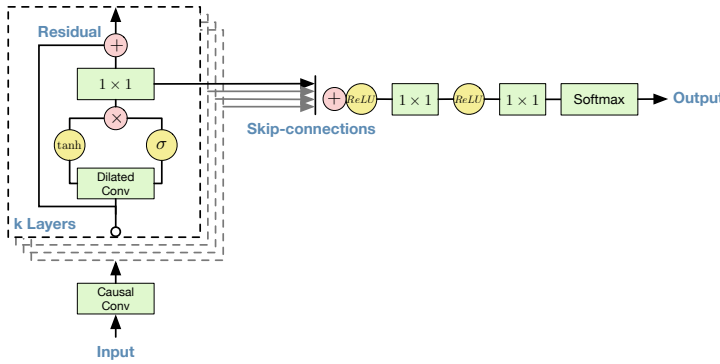

Figure 2.10. Overview of the residual block and the entire architecture, from [71].

These models can also be used as vocoder by using a mel-spectrogram as input instead of linguistic features. This allows to combine them with a different acoustic model, e.g., Sequence-to-sequence models.

Other approaches, such as Parallel Wavenet [72] and MelGAN [54], remove the autoregressive nature making them more viable solutions, since they are much more efficient at the time of inference. Parallel Wavenet distills a trained autoregressive Wavenet model into a flow-based convolutional student model. The student is trained using a probability distillation objective based on the Kulback-Leibler divergence (see equation 2.13). MelGAN, however, is a non-autoregressive model that does not need distillation of an autoregressive model. It uses a GAN setup and a fully convolutional architecture.

Sequence-to-sequence-based TTS Sequence-to-sequence modeling, presented earlier in this section, has been successfully applied to TTS. It is used with an attention mechanism that allows to improve the alignment of text and audio information, previously handled by Dynamic Time Warping (DTW), HMMs or duration model with DNNs.

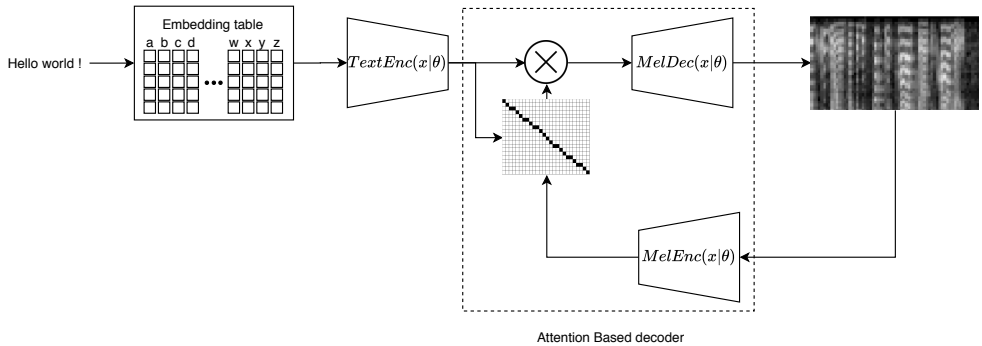

Figure 2.11. Block diagram of a typical DL-based TTS system.

Figure 2.11 represents a typical seq2seq TTS system. First the characters of the text are encoded in embedding vectors thanks to an embedding table. An embedding table is a table that associates a vector to an entry, i.e., a character in this case. The content of these vectors are parameters of the models such as the weight and biases of the networks. These are thus fine-tuned during training of the architecture. The text sentence is thus represented as a sequence of embedding vectors stored in a matrix. This matrix is then passed through an encoder called $TextEnc$. An Attention-based decoder is then used to predict a mel-spectrogram from this representation of text.

The Attention-based decoder is implemented as a causal process, i.e., it predicts a mel-spectrogram frame based on the alignment of text information and only past mel-spectrogram frames. This attention matrix is used as a weighting applied on the text representation at the output of $TextEnc$. The result is then passed through $MelDec$ that predicts mel-spectrogram frames.

DCTTS Most recent TTS systems, such as Wavenet [71], Tacotron [111], WaveRNN [46], Char2Wav [91] and Deep Voice [3], achieve excellent results in terms of naturalness. However they require tens of hours of speech data and a lot of computational power. A first system that aims to synthesize speech with few computational power is DCTTS [93]. This system is only based on CNNs and avoids using RNNs known to be difficult to train due to the van-

ishing gradient issue during gradient descent [41]. In their experiments, the authors of DCTTS were able to train their model in 15 hours using a standard computer with two GPUs, resulting in nearly acceptable speech synthesis.

They perform a MOS test to compare DCTTS model to the open source version of Tacotron[4] and obtain respectively 2.61 ± 0.62 and 2.07 ± 0.62. It is important to note that both methods use a Griffin-Lim algorithm as a Vocoder which gives lower MOS than recent neural vocoders [54].

As most of the experiments rely on DCTTS, the details of the different blocks are given in Figure 2.12. We use the notations introduced in [93] in which the reader can find more details if needed:

- 1D convolution: $\mathsf{C}_{k\star\delta}^{o\leftarrow i}(X)$, where i is the sizes of input channel, o is the sizes of output channel, k is the size of kernel, δ is the dilation factor, and an argument X is a tensor having three dimensions (*batch, channel, temporal*). The stride is 1.
- 1D *deconvolution*: layer as $\mathsf{D}_{k\star\delta}^{o\leftarrow i}(X)$. The stride is 2.
- Layer composition operator: $\cdot \triangleleft \cdot$, and
- Networks : $\mathsf{F} \triangleleft \mathsf{ReLU} \triangleleft \mathsf{G}(X) := \mathsf{F}(\mathsf{ReLU}(\mathsf{G}(X)))$, and $(\mathsf{F} \triangleleft \mathsf{G})^2(X) := \mathsf{F} \triangleleft \mathsf{G} \triangleleft \mathsf{F} \triangleleft \mathsf{G}(X)$, etc.
- ReLU is an element-wise activation function defined by $\mathsf{ReLU}(x) = \max(x, 0)$.
- Highway Network-like gated activation, $\mathsf{Highway}(X; \mathsf{L}) = \sigma(H_1) \odot H_2 + (1 - \sigma(H_1)) \odot X$,

  where H_1, H_2 are output by a layer L as $[H_1, H_2] = \mathsf{L}(X)$. The operator $\odot$ is the element-wise multiplication

  $\mathsf{HC}_{k\star\delta}^{d\leftarrow d}(X) := \mathsf{Highway}(X; \mathsf{C}_{k\star\delta}^{2d\leftarrow d})$.

[4] https://github.com/keithito/tacotron

$(K, V) = \mathsf{TextEnc}(L) := (\mathsf{HC}^{2d\leftarrow 2d}_{1\star 1})^2 \triangleleft (\mathsf{HC}^{2d\leftarrow 2d}_{3\star 1})^2 \triangleleft (\mathsf{HC}^{2d\leftarrow 2d}_{3\star 27} \triangleleft \mathsf{HC}^{2d\leftarrow 2d}_{3\star 9} \triangleleft \mathsf{HC}^{2d\leftarrow 2d}_{3\star 3} \triangleleft \mathsf{HC}^{2d\leftarrow 2d}_{3\star 1})^2 \triangleleft \mathsf{C}^{2d\leftarrow 2d}_{1\star 1} \triangleleft \mathsf{ReLU} \triangleleft \mathsf{C}^{2d\leftarrow e}_{1\star 1} \triangleleft \mathsf{CharEmbed}^{e-\dim}(L)$.

$Q = \mathsf{MelEnc}(S) := (\mathsf{HC}^{d\leftarrow d}_{3\star 3})^2 \triangleleft (\mathsf{HC}^{d\leftarrow d}_{3\star 27} \triangleleft \mathsf{HC}^{d\leftarrow d}_{3\star 9} \triangleleft \mathsf{HC}^{d\leftarrow d}_{3\star 3} \triangleleft \mathsf{HC}^{d\leftarrow d}_{3\star 1})^2 \triangleleft \mathsf{C}^{d\leftarrow d}_{1\star 1} \triangleleft \mathsf{ReLU} \triangleleft \mathsf{C}^{d\leftarrow d}_{1\star 1} \triangleleft \mathsf{ReLU} \triangleleft \mathsf{C}^{d\leftarrow F}_{1\star 1}(S)$.

$A = \mathsf{softmax}_{n-\mathrm{axis}}(K^{\mathsf{T}} Q/\sqrt{d}) \qquad R = \mathsf{Att}(Q, K, V) := VA \qquad R' = [R, Q]$

$Y = \mathsf{MelDec}(R') := \sigma \triangleleft \mathsf{C}^{F\leftarrow d}_{1\star 1} \triangleleft (\mathsf{ReLU} \triangleleft \mathsf{C}^{d\leftarrow d}_{1\star 1})^3 \triangleleft (\mathsf{HC}^{d\leftarrow d}_{3\star 1})^2 \triangleleft (\mathsf{HC}^{d\leftarrow d}_{3\star 27} \triangleleft \mathsf{HC}^{d\leftarrow d}_{3\star 9} \triangleleft \mathsf{HC}^{d\leftarrow d}_{3\star 3} \triangleleft \mathsf{HC}^{d\leftarrow d}_{3\star 1}) \triangleleft \mathsf{C}^{d\leftarrow 2d}_{1\star 1}(R')$.

$\mathsf{SSRN}(Y) := \sigma \triangleleft \mathsf{C}^{F'\leftarrow F'}_{1\star 1} \triangleleft (\mathsf{ReLU} \triangleleft \mathsf{C}^{F'\leftarrow F'}_{1\star 1})^2 \triangleleft \mathsf{C}^{F'\leftarrow 2c}_{1\star 1} \triangleleft (\mathsf{HC}^{2c\leftarrow 2c}_{3\star 1})^2 \triangleleft \mathsf{C}^{2c\leftarrow c}_{1\star 1} \triangleleft (\mathsf{HC}^{c\leftarrow c}_{3\star 3} \triangleleft \mathsf{HC}^{c\leftarrow c}_{3\star 1} \triangleleft \mathsf{D}^{c\leftarrow c}_{2\star 1})^2 \triangleleft (\mathsf{HC}^{c\leftarrow c}_{3\star 3} \triangleleft \mathsf{HC}^{c\leftarrow c}_{3\star 1}) \triangleleft \mathsf{C}^{c\leftarrow F}_{1\star 1}(Y)$.

Figure 2.12. Details of DCTTS architecture [93]

2.4.4 Information Theory and Speech Probability distributions

The previous section described different operations of Deep Learning architectures to construct models for TTS. These models have to be trained by minimizing a loss function between predictions of the model and examples from the dataset. These loss functions are based on concepts coming from the field of Information Theory.

Information Theory is about optimizing how to send messages with as few resources as possible. To that end, the goal is to compress the information by using the right code so that the messages do not contain redundancies to be as small as possible.

Information and probabilities

Shannon's Information Theory [62] quantifies information thanks to the probability of outcomes. If we know an event will occur, its occurrence gives no information. The less likely it is to happen, the more it gives information.

This relationship between information and probability of an event is given by Shannon information content measured in bits. A bit is a variable that can have two different values: 0 or 1.

$$h(x) = \log_2 \left(\frac{1}{p(x)} \right) \tag{2.9}$$

The number of possible messages with L bits is 2^L. If all messages are equally probable, the probability of each message is $p = \frac{1}{2^L}$. We then have $L = \log_2 \left(\frac{1}{p} \right)$. A generalization of this formula in which the messages are not equally probable is Equation 2.9. It can be interpreted as the minimal number of bits to communicate this message.

The probability represents the degree of belief that an event will happen [62]. For example, we can wonder the probability of a result of four by rolling a six sided die or the probability that the next letter in a text will be a r.

These probabilities depend on the assumptions we make:

- Is the die perfectly balanced? If yes, the probability of a result of four is 1/6.
- What is the language of the text ? Do we know the subject, etc. Depending on this information we can have different estimations of this probability.

We obtain a probability distribution by listing the probability of all the possible outcomes. For the example of the result by rolling the perfectly balanced die, the possible outcomes are $[1, 2, 3, 4, 5, 6]$ and their probabilities are $[1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$.

In both examples, we have a finite number of possible outcomes. The probability distribution is said to be discrete. On the contrary, when the possible outcomes are distributed on a continuous interval, then the probability distribution is said to be continuous. This is the case, for example, of amplitude values in a spectrogram.

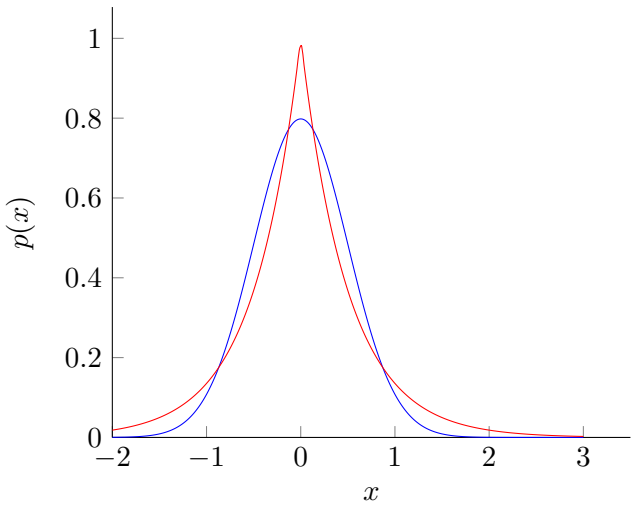

Figure 2.13. Example of widely used probability distributions. In blue: Gaussian distribution with $\mu = 0$ and $\sigma = 0.5$. In Red: Laplacian distribution with $\mu = 0$ and $b = 0.5$.

The most famous continuous probability distribution is the Gaussian distribution:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2.10}$$

Another important distribution, especially in speech processing is Laplacian distribution:

$$p(x) = \frac{1}{2b} e^{\left(-\frac{|x-\mu|}{b}\right)} \tag{2.11}$$

Both distributions are plotted in Figure 2.13. The blue curve corresponds to the Gaussian probability distribution (with $\mu = 0$ and $\sigma = 0.5$) and the red curve corresponds to the Laplacian probability distribution (with $\mu = 0$ and $b = 0.5$). For both distributions, the maximum is the mean μ. They are symmetrically decreasing as the distance from μ increases.

Entropy and relative-entropy

The average information content of an outcome, also called entropy, of the probability distribution p is:

$$H(p) = \sum_x p(x) \log_2 \left(\frac{1}{p(x)} \right) \tag{2.12}$$

The relative-entropy between two probability distributions, also called Kullback-Leibler divergence is defined as:

$$D_{\mathrm{KL}}(p|q) = \sum_x p(x) \log \frac{p(x)}{q(x)} dx \tag{2.13}$$

It represents a dissimilarity between two probability distributions.

Maximum likelihood and particular cases

This concept is necessary to understand how to train a Deep Learning algorithm or more generally, how to find the optimal model parameters. The role of a statistical model is to represent as accurately as possible the behaviour of a probability distribution.

Maximum Likelihood Estimation (MLE) (see Equation 2.14) allows the estimation of the parameters θ of a statistical parametric model $p(x|\theta)$ by maximizing the probability of a dataset under the assumed statistical model, i.e., the Deep Learning architecture.

$$\theta_{MLE} = \arg\max p(\mathbf{x}|\theta) \tag{2.14}$$

It can be demonstrated this is equivalent to minimizing $D_{\mathrm{KL}}(p|q)$ with p, the probability distribution of the model and q, the probability of the data [36]. It is a way to express that the probability distribution generated by the model

should be as close as possible to the probability distribution of the data.

If assumptions can be made on the probability distributions, it is possible to have distances or errors for which the minimization is equivalent to MLE. These errors are computed by comparing estimations from the model $\hat{Y}_i$ and the value from the dataset Y_i.

Maximizing likelihood assuming a Gaussian distribution is equivalent to minimizing Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.15}$$

Maximizing likelihood assuming a Laplacian distribution is equivalent to minimizing Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{2.16}$$

To choose the right criterium to optimize when working with speech data, one should pay attention to speech probability distributions. Speech waveforms and magnitude spectrogram distribution are Laplacian [35, 108]. That is why MAE loss should be used to optimize their predictions.

2.5 Summary and Application

In this chapter, we first briefly introduced digital signal processing and digital filtering, and described the different possibilities of emotion representation and the few most important speech feature spaces in this context, namely spectrogram and mel-spectrogram.

Available speech synthesis methods were then exposed: concatenation of speech signal segments, modeling of speech production and statistical parametric speech synthesis. Depending on the synthesis technique used [10], the voice is more or less natural and the control parameters are more or less numerous. These parameters allow the creation of variations in the acoustic features of the voice. The number of parameters is therefore important for the synthesis of expressive speech. While Source-Filter-based speech synthesis gives access to many control parameters, the resulting voice is unnatural. Synthesizers using the principle of concatenation of speech segments seem more natural but allow the control of few acoustic features. The statistical approaches lead to a natural synthesis as well as a control of many acoustic features [123].

Most recent SPSS systems use Deep Learning that can be seen as non-linear signal processing for which filters are optimized based on data. We focused on the tools for SPSS and explained Deep Learning architecture blocks that are used along with the right loss functions based on the probability distributions of speech features.

To build a controllable expressive speech synthesis system, one should keep several concepts in mind. First, it is necessary to gather data and process them to have a good representation to be used with a Deep Learning algorithm, i.e., text, mel-spectrograms, and information about the expressiveness of speech. Then one has to design a Deep Learning architecture. Its operations should be inspired by the features to model (1D convolution or RNN cells for long term context, attention mechanism for recursive relationships). It should have a way to control expressiveness either with a categorical representation or a continuous representation. And finally, the model should be trained with a loss function adapted to the probability distribution of the acoustic features, i.e., MAE and Kullback-Leibler divergence loss.

Part II

Experiments

Chapter 3

Speech Datasets

Contents

This chapter is based on the following publication and abstract (respectively):

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Emotional Speech Datasets for English Speech Synthesis Purpose: A Review". In: *Intelligent Systems and Applications*. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 61–66. ISBN: 978-3-030-29516-5
- Adaeze Adigwe, Tits Noé, El Haddad Kevin, Ostadabbas Sarah, Dutoit Thierry, "The Emotional Voices Database: Towards Controlling the Emotion Dimension in Voice Generation Systems". In: *"International Conference on Statistical Language and Speech Processing"*, Mons, Belgique (2018)

This chapter is part of the first of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., review and collect neutral and emotional speech data that is necessary for the other tasks.

We review the datasets of emotional speech publicly available and their usability for state of the art speech synthesis. This is conditioned by several characteristics of these datasets: the quality of the recordings, the quantity of the data and the emotional content contained in the data. The description of these datasets is essential for the rest of the thesis.

We then present a dataset, named EmoV-DB, that was recorded based on the observation of the needs in this area. It contains data for male and female actors in English and a male actor in French. The database covers 5 emotion classes so it could be suitable to build synthesis and voice transformation systems with the potential to control the emotional dimension.

A voice conversion experiment is conducted on EmoV-DB. The aim is to show that the quality of the recordings is sufficient for applications in the field of voice generation with emotions.

3.1 Introduction

One of the major components of human-agent interaction systems is the speech synthesis module. The state-of-the-art speech synthesis systems such as wavenet and tacotron [71, 111, 89] are giving impressive results. They can produce, intelligible, expressive, even human-like speech. But, they cannot yet be used to control the emotional dimensionality in speech which is a crucial aspect in order to obtain a human-like controllable speech synthesis system.

Although still being relatively neglected by the affective computing community, the interest for emotional speech synthesis systems has been growing for the past two decades. After the improvement parametric systems brought to this field [26], deep learning-based systems were also employed for such a task.

One of the problems in the emotional speech synthesis research community is the lack of open-source data available and the difficulty to collect them. In fact, to the best of our knowledge, no open-source emotional speech database for synthesis purpose and suitable for deep learning systems is available. In this chapter, we try to tackle this problem. We present an open-source multi-speaker (5 different speakers) and multilingual (in North American English and Belgian French) database of emotional speech. The database contains enough data of good quality to train deep learning-based systems for speech control and generation applications. We thus propose an emotional speech dataset version of the CMU-Arctic speech database [51] which has been one of the reference open-source databases for speech synthesis since the early 2000.

In what follows we will present the background introducing this work in Section 3.2. We will then detail the motivation for data collection Section 3.3 and detail the content of our database in Section 3.3.1. In order to validate our database we will use it in a voice transformation experiment. Indeed the data will be used to transform neutral to emotional voice using deep learning-based systems and the obtained results will be evaluated using a Comparative Mean Opinion Score test in Section 3.3.2.

3.2 Background

3.2.1 Open-source recorded datasets

Several open-source databases can be found but to the best of our knowledge, none is really suitable for a purpose of emotional speech synthesis. In this section we will explain why and mention some examples.

The RAVDESS database contains emotional data from 24 different actors [60]. The actors were asked to read 2 different sentences in a spoken and sung way in North American English. The spoken style was recorded in 8 different emotional styles: neutral, calm, happy, sad, angry, fearful, disgust, surprise. Each utterance was expressed at 2 different intensities (except for the neutral emotion) and 2 times thus giving a total of 1440 files. A perception test was then

undertaken to validate the database on the emotional categories, intensity and genuineness.

The CREMA-D database [15] is similar to the RAVDESS. For this database, 12 different sentences were recorded by 91 different actors, for the 6 basic emotions: happy, sad, anger, fear, disgust, and neutral. Only one of the 12 sentences was expressed in 3 different intensities, for the other 11, the intensity was not specified. The authors report 7442 files in total. This database was also validated through perception tests and helped validate the emotion category and intensity.

Also similar to the previous ones, the GEMEP database [4] is a collection of 10 French-speaking actors, recorded uttering 15 different emotional expressions at three levels of intensity, in three different ways: improvised sentences, pseudo-speech, and nonverbal affect bursts. This database counts a total of 1260 audio files. It was also validated through perception tests.

The Berlin Emotional Speech Dataset [11] contains the recording of 10 different utterances by 10 different actors in 7 different emotions (neutral, anger, fear, joy, sadness, disgust and boredom) in German, making it a total of 800 utterances (counting some second version of some of the sentences). This database was, like the previous ones, validated using perception experiments.

These databases are not suitable for current state of the art speech synthesis purposes because of the limited amount of sentences recorded.

Moreover, the six basic emotions do not really occur in daily conversations. Indeed, in Ekman's model, on which the choice of emotions was based for these datasets, the basic emotions are the ones from which other emotions derive. But that does not necessarily mean that they are frequently expressed in speech in our daily interactions.

The IMPROV [14] and IEMOCAP [13] databases both contain a large amount of diverse sentences of emotional data. IEMOCAP contains audio-visual record-

ings of 5 sessions of dyadic conversations between a male and a female subject. In total it contains 10 speakers and 12 hours of data. IMPROV contains 6 sessions from 12 actors resulting in 9 hours of audiovisual data. Both databases were evaluated in terms of category of emotions [23] and emotional dimensions [80] by several subjects. However they are not suitable for a synthesis purpose either because although the data is well recorded and post-processed it contains overlapping speech due to the data recording setup (dyadic conversation) and some external noise.

The CMU Arctic Speech Database [51] and the SIWIS French Speech Synthesis Database [42] are collections of read utterances of phonetically balanced sentences in English and French respectively. The CMU-Arctic database contains approximately 1150 sentences recorded from each of 4 different speakers while SIWIS contains a total of 9750 utterances from a single speaker. These are databases suitable for a speech synthesis purpose as there is a large amount of different sentences recorded from a single speaker in a noiseless environment. However the sentences are neutral and do not express any emotions.

The AmuS database contains audio data dedicated to amused speech synthesis [30]. We showed in previous work [24, 25, 27] that this database was well suited for amused speech synthesis. But AmuS contains data only for amused speech and not other emotions. This dataset contains laughters with a dedicated annotation scheme. This is described in more details in Chapter 6 dedicated to acoustic laughter synthesis.

3.2.2 Proprietary dataset

Some of the experiments of this project were done with a dataset provided by ACAPELA GROUP SA. The goal of these recordings was to build a storytelling system to build audiobooks from transcripts.

It contains phonetically rich sentences uttered by a male actor in English. The actor was asked to utter a set of the sentences in 8 style classes. For the sake of clarity, we will refer to Will in the following sections to designate the actor.

The instructions/examples given to Will to speak with different styles are the following:

- neutral: typical narration;
- happy: smile and positive;
- sad: depressed;
- bad guy: mean;
- from afar: "open the gate!" said the knight;
- proxy: "don't make too much noise or the monster will hear you", whispering;
- old man: mimic an old man's voice;
- little creature: little monster.

For each sentence, there is a wave file sampled at 22.05 kHz and coded in 16 bit linear and a corresponding transcription. The duration of audio files are given in Table 3.1 in minutes. The durations after trimming silences are also indicated.

Table 3.1. Durations (min), duration after trimming silences (min) an number of utterances for each style.

	Duration	Trimmed duration	n utts
NEUTRAL	240.68	150.50	3299
HAPPY	152.00	97.08	2130
SAD	199.02	142.20	2130
BADGUY	179.74	113.57	1867
FROMAFAR	190.02	119.14	2130
PROXY	179.37	123.86	2232
OLDMAN	239.51	134.38	2130
LITTLECREATURE	214.41	124.14	2156

3.2.3 Audiobook based datasets

The LJ Speech Dataset [44] is a large public domain speech dataset containing recordings of one female speaker. This dataset has become a standard to evaluate the performance of different speech synthesis systems. Statistics of the dataset are given online[1] and transcribed in Table 3.2.

Table 3.2. Statistics of LJ Speech Dataset.

Total Clips	13,100
Total Words	225,715
Total Characters	1,308,678
Total Duration	23:55:17
Mean Clip Duration	6.57 sec
Min Clip Duration	1.11 sec
Max Clip Duration	10.10 sec
Mean Words per Clip	17.23
Distinct Words	13,821

The scripts uttered by the speaker comes from seven non-fiction books. In terms of expressiveness, the dataset is not very diverse. It is therefore more suited for neutral TTS than for expressive TTS.

LibriTTS [121] is another large corpus suitable for TTS. It is a curated and post-processed version of LibriSpeech dataset that was mostly used for training and evaluation speech recognition systems.

Blizzard 2013 dataset[2] is, like LJ Speech, composed of recordings of a single female speaker. It is however a lot more expressive than the latter.

[1] https://keithito.com/LJ-Speech-Dataset/
[2] http://www.cstr.ed.ac.uk/projects/blizzard/2013/lessac_blizzard2013/

3.3 EmoV-DB

The EmoV-DB's[3] primary purpose is to build models that could not only produce emotional speech but also control the emotional dimension in speech.The techniques to allow this are either TTS like systems where the system would map a given text sentence to a speech audio signal or voice transformation systems where a source voice would be converted to a specific target emotional voice. Considering this, it is obvious that a lot of data is required. One of the primary difficulties of building emotional speech-based generation systems is the collection of data. Indeed not only must the recording be of good quality and noise free, but the task of expressing emotional sentences in a large enough amount is challenging. Also, it is often preferable concerning these types of systems, that a certain category of emotion contains data that are similar on the acoustic level.

The database presented here was built with these requirements in mind. The aim was also for it to fit with other currently open-source databases to maximize the quantity of data available. As mentioned previously, the CMU-Arctic database (English) and the SIWIS (French) databases are two datasets of neutral speech. Each of them contains a relatively large amount of data that can be used as source voices for a voice conversion system or as pre-training data for a system. They are also transcribed which makes the transcription also available for our database. The transcribed utterances as well as annotations at phonetic level are available. A subset of these were used to build our database. The phonetic annotations are not time-aligned with our data yet, but methods can be used such as forced alignment systems [9].

We chose five different emotions: amusement, anger, sleepiness, disgust and neutral. We chose emotions that are more likely to be expressed in daily conversations than Ekman's basic emotions. These emotions were chosen because of the ease to produce them by actors and in order to cover a diverse space in the Russel Circumplex to allow experimenting with interpolation techniques to obtain intermediate emotions.

[3] https://github.com/numediart/EmoV-DB

Table 3.3. Repartition of the sentences of EmoV-DB dataset by gender, language and emotion.

Speaker	Gender	Language	Neutral	Amused	Angry	Sleepy	Disgust
Spk-Je	Female	English	417	222	523	466	189
Spk-Bea	Female	English	373	309	317	520	347
Spk-Sa	Male	English	493	501	468	495	497
Spk-Jsh	Male	English	302	298	-	263	-
Spk-No	Male	French	317	-	273	-	-

3.3.1 Database Content

The data was recorded in 2 different languages English (North American) and French (Belgian). English natives (2 females and 2 males) and a single male French native were asked to read sentences while expressing one of the above mentioned emotions. The English sentences were taken from the CMU-arctic database. The French ones from the SIWIS database. Both databases contain freely available open-source phonetically balanced sentences.

The recordings for the English data were carried on in two different anechoic chambers of the Northeastern University campus. The ones for the French data were made in an anechoic room at the University of Mons.

The utterances were recorded in several sessions of about 30 minutes recordings followed by a 5 to 15 minutes break and the data collection was spread across several days depending on the availability of the actors. The actors were asked to repeat sentences that were mispronounced.

The actors were asked to record each emotion class separately in different sessions. The sentences were segmented manually for some of the speakers. By segmentation we mean determining the intervals of start and end of each sentence. The total number of utterances obtained is summarized in Table 3.3.

Amused speech can contain chuckling sounds which overlap and/or intermingle with speech called speech-laughs [106] or can be only amused smiled speech [26]. Therefore, for the amused data in our database, in order to collect as much data as possible and considering the relatively limited time the actors provided us, we focused on amused speech with speech-laughs. This choice was motivated by our previous study showing that this type of amused speech was perceived as more amused than amused smiled speech (without speech-laugh). Also in another study, we show that including laughter in synthesized speech is always perceived as amused no matter the style of speech it is inserted in (neutral or smiled) [27]. Based on the previous studies made on amusement, the actors were encouraged, while simulating the other emotions, to use nonverbal expressions [28] before and even while uttering the sentences if they felt the need to (e.g., yawning for sleepiness, affect bursts for anger and disgust).

3.3.2 Data Validation in a Voice Transformation Experiment

In order to validate our database, we show the performance of the data in a voice transformation system intended to generate target emotional speech from a source sentence. We thus designed an experiment described in this section.

Voice Transformation System

A feedforward-based voice transformation system was trained per speaker and per emotion. This experiment concerns Spk-Bea (female-English), Spk-Sa (male-English) and Spk-No (male-French). The system was trained to transform the neutral style (source) to another emotion style target. In this study, the target speech style is anger. This choice was motivated by the fact that the anger class was the one with the least nonverbal expressions which made the transformation task less complex for the simple system we chose. For each of the speakers a system was also trained to transform from neutral to neutral speech style. This was done to use the generated neutral utterances as reference and compare the emotional utterances generated to it. This thus gave us 6 systems trained in total.

Table 3.4. Amount of data used for training two Voice Conversion systems to experiment the usefulness of the EmoV-DB dataset. The first is a baseline that goes from neutral to neutral and the second goes from neutral to angry categories.

Pairs	Spk-Bea	Spk-Sa	Spk-No
neutral-neutral	355	456	243
neutral-angry	296	456	243

The voice transformation system is based on the Merlin Toolkit [116]. This toolkit contains a module allowing Voice Conversion (VC), i.e., transform a source speaker's voice so that it sounds like a target speaker's voice. The VC module do this by extracting speech features with a vocoder (the default WORLD vocoder [68] was used) of both source and target voices, performing a DTW to align the features in time and computing a regression between the source and target features. The regression model used is a simple DNN of 6 feedforward hidden layers in which each hidden layer is constituted of 1024 hyperbolic tangent units.

In this experiment, instead of training the VC module with sentences uttered by a source and a target speaker, we trained it with sentences uttered by the same speaker with a source and a target emotion category. The procedure of this experiment is the same as in [61] which showed good results for emotion to neutral speech transformations.

Table 3.4 shows the amount of training data used to train each system.

Perception Test

After training, 5 neutral test sentences from the recorded data, not seen previously by the systems during training in each case, were transformed by each system to its target emotion. Then, each source-target pair was used in a

Table 3.5. Percentage of angry and neutral speech styles being accurately classified in the listening test.

Pair	Spk-Bea	Spk-Sa	Spk-No
neutral-neutral	96%	90%	98%
neutral-angry	78%	71%	83%

Table 3.6. Mean and standard deviation of results obtained. Negative values would correspond to neutral being perceived as more emotional than the "anger" utterance, and vice versa for the positive values. A "0" grade would indicate that there is no difference between the compared utterances.

Pair	Spk-Bea mean/std	Spk-Sa mean/std	Spk-No mean/std
neutral-neutral	0.05/0.2	0.05/0.1	0.01/0.09
neutral-angry	2.3/1.2	2/2	2.4/1.3

Comparative Mean Opinion Score (CMOS) test. This makes it 30 different input-output pairs (including the neutral-neutral transformations).

Each pair was formed by the output of the system generating the neutral style with itself (neutral-neutral pair) or with the output of a system generating the angry style (neutral-angry pair) for the same speaker. Audio files were then created by concatenating both utterances to be compared in a single file with a 3 second silence delay between them. The order by which they were concatenated was random.

During this test, 26 participants per speaker were asked to grade on a scale of integers from -3 to 3 (0 included) which sentence was more emotional. They were then asked to pick among a list of 6 categories (neutral, sleepiness, anger, amusement, disgust or other) which one represented the most emotional utterance (0 corresponding to "no difference").

The more positive the value was, the more sure the participant considered that the first utterance in the audio clip was more emotional than the second utterance (and vice-versa for the negative). When processing the data, all ratings estimating that the neutral style was more emotional was converted to negative values and all others to positive. In case of the neutral-neutral pair all the ratings were therefore negative.

Table 3.5 shows the percentage of times the emotional utterances (utterance with highest grade or both if 0) were correctly categorized in each pair of speech styles. We can observe that accuracies for neutral to angry are lower than neutral to neutral. This is because the task is more challenging. The mean and standard deviation of the scores obtained by each pair are given in Table 3.6.

We can see from the above tables that the participants could recognize the angry expression accurately and with high confidence for each speaker. These results show that the data for the "anger" emotion, can efficiently be used for a voice transformation system.

It is interesting to note that most of the misclassification of the results in both the neutral-neutral case and the neutral-angry case were due to perceiving the neutral (or one of the neutrals) as sleepiness or amusement. Indeed the test being a comparative one these misclassifications might be due to classifying one expression with respect to the other instead of in an absolute way.

3.4 Conclusions

In this chapter, we reviewed existing datasets containing emotional speech and detailed their characteristics that condition the tasks for which they can be useful. We propose a first step towards obtaining a large open-source database of emotional data dedicated to systems aiming at controlling the emotional dimension in speech. We showed that the proposed database was efficient to produce angry voices from neutral ones using a simple DNN.

Chapter 4

Transfer Learning for Emotion recognition

Contents

This chapter is based on the following publication:

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "ASR-based Features for Emotion Recognition: A Transfer Learning Approach". In: *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 48–52. URL: http://aclweb.org/anthology/W18-3307

This chapter is part of the second of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., study the possibility to extract a representation of emotional expressiveness in speech with a Deep Learning architecture.

In this chapter, we investigate the use of a neural ASR as a feature extractor for emotion recognition. We show that these features outperform the eGeMAPS feature set to predict the valence and arousal emotional dimensions, which means that the audio-to-text mapping learned by the ASR system contains information related to the emotional dimensions in spontaneous speech. We also examine the relationship between first layers (closer to speech) and last layers (closer to text) of the ASR and valence/arousal.

The idea of using a Deep Learning architecture trained on a task to extract a representation of emotional expressiveness is used in Chapter 8.

4.1 Introduction

With the advent of deep learning, areas of signal processing have been drastically improved. In the field of speech synthesis, Wavenet [71], a deep neural network for generating raw audio waveforms, outperforms all previous approaches in terms of naturalness. One of the remaining challenges in speech synthesis is to control its emotional dimension (happiness, sadness, amusement, etc.). The work described in this chapter is a first step towards the control of the emotional state of a sentence being synthesized. For this, we present here exploratory work regarding the analysis of the relationship between the emotional states and the modalities used to express them in speech.

Indeed, one of the main problems to develop such a system is the amount of good quality data (naturalistic emotional speech of synthesis quality, i.e., containing no noise of any sorts). This is why we are considering solutions such as synthesis by analysis and transfer learning [74].

Arousal and valence [86] are among the most, if not the most used dimensions for quantizing emotions. Valence represents the positivity of the emo-

tion whereas arousal represents its activation. Since they represent emotional states, these dimensions are linked to several modalities that we use to express emotions (audio, text, facial expressions, etc.).

It has recently been shown that for emotion recognition, deep learning based systems learn features that outperform handcrafted features [105, 63, 48, 49]. The use of context and different modalities has also been studied with deep learning models. [79] focus on the contextual information among utterances in a video while [119, 120] develop specific architectures to fuse information coming from different modalities.

In this work, with the goal to study the relationship between valence/arousal, and different modalities, we propose to use the internal representation of a speech-to-text system. An ASR system or speech-to-text system, learns a mapping between two modalities: an audio speech signal and its corresponding transcription. We hypothesize that such a system must also be learning representations of emotional expressions since these are contained intrinsically in both speech (variation or the pitch, the energy, etc.) and text (semantic of the words).

We show here that the activations of certain neurons in an ASR system are useful to estimate the arousal and valence dimensions of an audio speech signal. In other words, transfer learning (see Section 2.4.3) is leveraged by using features learned for an ASR task to estimate valence and arousal. The advantage of our method is that it allows combining the use of large datasets of speech with transcriptions with limited datasets annotated in emotional dimensions.

An example of transfer learning application in the field of affective computing is the work of [82]. The authors trained a multiplicative LSTM [52] to predict the next text character based on the previous ones to design a text generator system. The dataset used to train their model was the Amazon review dataset presented in [64]. Then, they used the representation learned by the model to predict sentiment also available in the dataset, and achieved state of the art

prediction.

In this chapter, we show that the activations of a deep learning-based ASR system trained on a large database can be used as features for the estimation of arousal and valence values. The features would therefore be extracted from both the audio and text modalities which the ASR system learned to map.

4.2 ASR-based Features for Emotion Prediction Via Regression

Our goal is to study the relationship between valence/arousal, and audio/text modalities thanks to an ASR system. The main idea is that the ASR system that models the mapping between audio and text might learn a representation of emotional expression. So, for our analyses, we use an ASR system as a feature extractor which feeds a linear regression algorithm to estimate the arousal/valence values. This section describes the whole system. First we present the ASR system used as a feature extractor. We then briefly present the data used and present first results on the data analysis.

4.2.1 ASR system

The ASR system used is implemented in [70] and pre-trained on the VCTK dataset [109] containing 44 hours of speech uttered by 109 native speakers of English.

Its architecture, depicted in Figure 4.1, consists of a dilated convolution of blocks. Each block is a GCU with a skip (residual) connection. In other words a Wavenet-like architecture [71] as described in Section 2.4.3. There are 15 layers and 128 GCUs in each layer: 1920 GCUs in total. To lighten the computational cost, the audio signal is compressed in 20 Mel Frequency Cepstral Coefficients (MFCCs) and then fed into the system.

The model predicts a sequence of ASCII characters that are compared to ground truth via a Connectionist Temporal Classification (CTC) loss [37] that

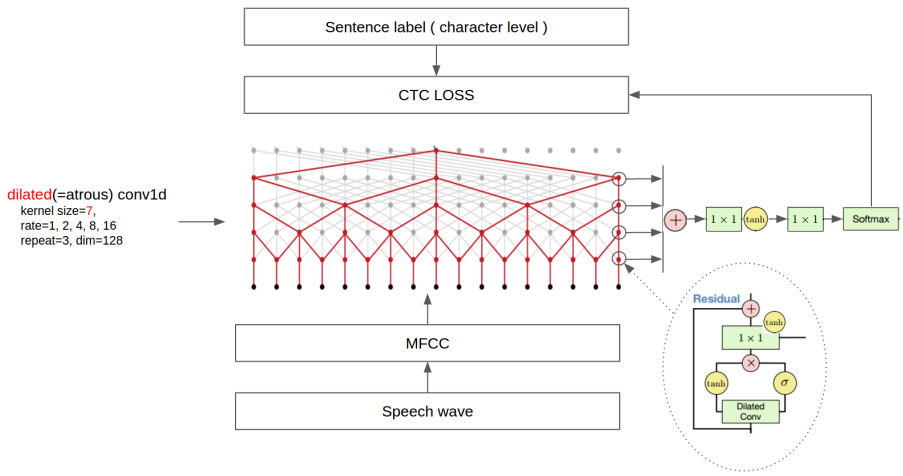

Figure 4.1. ASR Network architecture. [70]

allows using neural networks classifiers for temporal classification tasks, e.g., speech recognition.

The model is only an acoustic model. It could be augmented with a post-processing step by adding a statistical language model for better performance in terms of speech recognition. A language model's role is to correct text predictions with a model trained on text data. However in this context, we are not interested to further improve text predictions after the acoustic modeling step.

4.2.2 Dataset Used

IEMOCAP Dataset

The "interactive emotional dyadic motion capture database" (IEMOCAP) [13] is used in this chapter. It consists of audio-visual recordings of 5 sessions of dialogues between male and female subjects. In total it contains 10 speakers and a total of 12 hours of data. The data is segmented in utterances. Each

utterance is transcribed and annotated by category of emotions [22] and a value for emotional dimensions [86] (valence, arousal and dominance) between 1 and 5 representing the dimension's intensity. In this work, we only use the audio and text modalities as well as the valence and arousal annotations.

Data Analysis and Neural Features

We investigate the relationship between the activation output of the ASR-based system's GCUs and the valence/arousal values by studying the correlations between them. For every utterance and for each speaker of the IEMOCAP dataset, we compute the mean activation of the GCUs of the ASR. The Pearson correlation coefficient is then calculated between the mean activation outputs and the values of valence/arousal of all utterances of the speaker. In the rest of the chapter, we will refer to the mean activation of the GCUs as neural features. As an example, the results concerning the female speaker of session 2 is summarized in a heat map represented in Figure 4.2

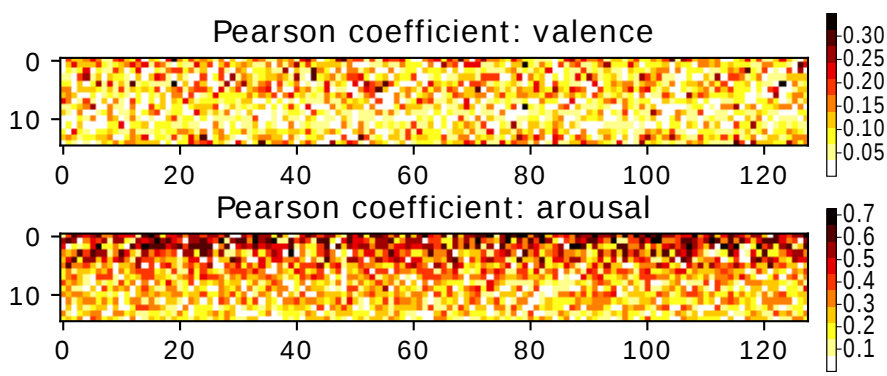

Figure 4.2. Absolute Pearson correlation coefficient between the neural features and valence (up) and arousal (down) - Female speaker of session 2. The neural features are obtained by averaging, for each utterance, the activation of each convolutional block in each layer of the architecture. In each heat map, lines correspond to the different layers and columns to the different convolutional blocks of these layers.

Each row of the heat map corresponds to a layer of GCUs. The color is mapped with the Pearson correlation coefficient value.

One can see that correlations exist for both arousal and valence. This suggests that the ASR-based system learns a certain representation of the emotional dimensions.

4.2.3 Structure of the system

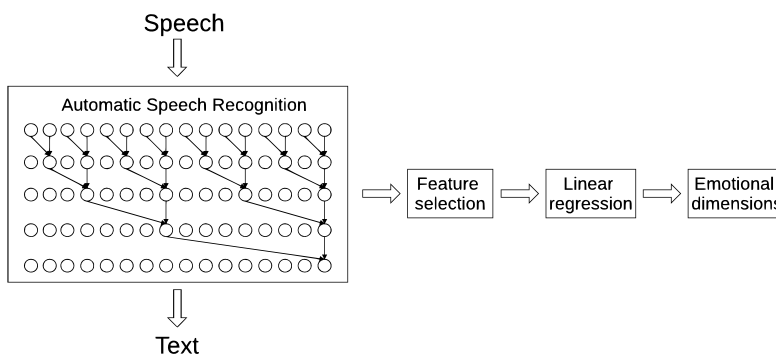

Figure 4.3. Block diagram of the system allowing the prediction of emotional information from a Deep Learning-based ASR system. A pre-trained ASR system taking MFCC features as input and predicting characters is used as a feature extractor. The features correspond to the output of sub-blocks of the architecture. Feature selection is applied using Fisher score as criterion. Valence and arousal are then predicted with a linear regression of the features.

The system is illustrated in Figure 4.3. As previously mentioned, the ASR system is used as a feature extractor. First we compute the 20 MFCCs of the utterances of the IEMOCAP dataset with librosa python library [65], a standard audio processing library. These are passed through the ASR to compute the corresponding neural features.

A feature selection is applied on the neural features to keep 100 among the 1920 features for dimensionality reduction purpose. The selection is done us-

ing the scikit-learn python library [76] with the Fisher score.

Finally a linear regression is trained to estimate the valence/arousal values from the neural features using the IEMOCAP data. The linear regression is done using scikit-learn. The training is done by minimizing the MSE between predictions and labels.

4.3 Experiments and Results

In this section, we detail the experiments that we carried out. The first one is the evaluation of the neural features in terms of MSE and its comparison with a linear regression of the eGeMAPS [33] feature set. In the second one, we investigate the relationship between the audio and text modalities and the emotional dimensions.

4.3.1 First experiment: Linear regression

In this first experiment, we investigate the performance of a linear regression to predict arousal and valence using the neural features. We compare this with a linear regression using the eGeMAPS feature set.

The eGeMAPS feature set is a selection of acoustic features that provide a common baseline for evaluation in researches to avoid differences of feature set and implementations. Indeed, an implementation is provided with openSMILE toolkit [34].

The eGeMAPS features were selected based on their ability to represent affective physiological nuances in voice production, their proven performance in former research work as well as the possibility to extract them automatically, and their theoretical significance.

The feature set is based on Low Level Descriptors (LLDs) (F0, formants, MFCCs, etc.) to which are applied statistics for the utterance (mean, nor-

malized standard deviation, percentiles). All statistics are applied to voiced regions only (non-zero F0). For MFCCs, there is also a version applied to all regions (voiced and unvoiced).

These features are defined in [33] as follows:

- F0: logarithmic F0 on a semitone frequency scale, starting at 27.5 Hz (semitone 0);
- F1-3: Formants 1 to 3 centre frequencies;
- Alpha Ratio: ratio of the summed energy from 50-1000 Hz and 1-5 kHz;
- Hammarberg Index: ratio of the strongest energy peak in the 0-2 kHz region to the strongest peak in the 2-5 kHz region;
- Spectral Slope 0-500 Hz and 500-1500 Hz: linear regression slope of the logarithmic power spectrum within the two given bands;
- mfcc1-4: Mel-Frequency Cepstral Coefficients 1 to 4.

The results obtained from the linear regression in terms of MSE are compared to the annotations for each of the arousal and valence values (between 1 and 5) in Table 4.1. We perform a leave-one-speaker-out evaluation scheme with both feature sets for cross-validation. In other words, each validation set is constituted with the utterances corresponding to one speaker and the corresponding training set with the other speakers. We train a model with each training set and evaluate it on the validation set in terms of MSE. The table contains the mean and standard deviation of the MSEs.

	Arousal		Valence	
	Mean	Variance	Mean	Variance
Neural features	**0.259**	0.020	**0.660**	0.118
eGeMAPS set	0.267	0.034	0.697	0.135

Table 4.1. Means and variance of the MSE (lower is better) on the prediction of valence and arousal by a linear regression trained on the eGeMAPS feature set and the neural features.

This table shows that the neural features outperform the eGeMAPS features in this experiment. This confirms the fact that the ASR system learns representations of emotional dimensions in spontaneous speech.

4.3.2 Second experiment: Influence of modalities

During the data exploration, we noticed that, for some speakers, the layers closer to the speech input were more correlated to arousal and the ones closer to the text output to valence. An example is shown in Figure 4.4. We present, in this section, preliminary studies regarding this matter.

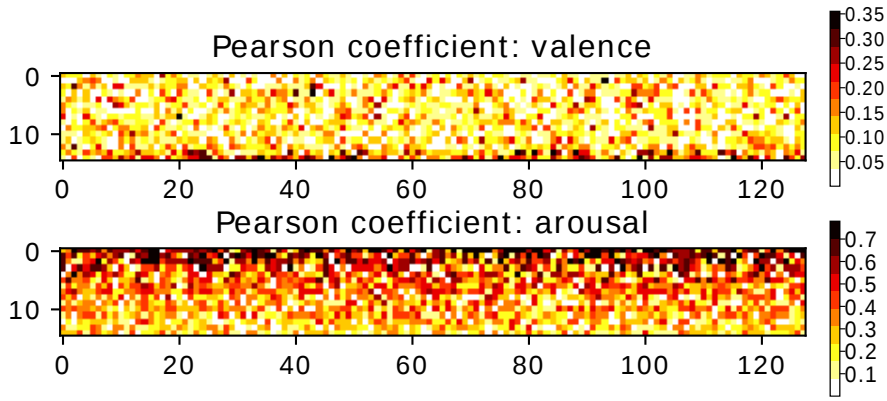

Figure 4.4. Pearson correlation coefficient between the neural features and valence (up) and arousal (down) - Female speaker of session 1.

In order to analyze this phenomenon as precisely as possible, we only considered the utterances from the IEMOCAP database for which the annotators reached an agreement between with each other. This was evaluated using categorical labels. In case of ex aequo in votes for the different categories, it was considered not consistent in terms of inter-evaluator agreement; 7532 segments out of the 10039 segments reached agreement.

Then we performed linear regression with 4 different sets of feature to study their influence. For the first set, we select the 100 best features among the 3

first layers of the neural ASR in terms of Fisher score using scikit-learn. For the second set, we apply the same selection to the 3 last layers. The third set selection is applied among all neural features. The last set is the eGeMAPS feature set.

The results are summarized in Table 4.2. As expected, the results show, that for the speakers considered, the layers closer to the audio modality outperform the ones closer to the text modality in the ASR architecture for arousal prediction and vice versa for the valence prediction.

	Arousal		Valence	
	Mean	Variance	Mean	Variance
First layers	0.325	0.069	0.714	0.114
Last layers	0.357	0.038	0.661	0.089
All	0.296	0.044	0.621	0.099
eGeMAPS set	0.328	0.064	0.683	0.124

Table 4.2. Means and variances of the MSE on the prediction of valence and arousal.

4.4 Conclusions

In this chapter, we show that features learned by a deep learning-based system trained for the Automatic Speech Recognition task can be used for emotion recognition and outperform the eGeMAPS feature set, the state of the art handcrafted features for emotion recognition. Then we investigate the correlation of the emotional dimensions arousal and valence with the modalities of audio and text of the speech. We show that for some speakers, arousal is more correlated to neural features extracted from layers closer to the speech modality and valence to the ones closer to the text modality.

From these results, we also notice that valence remains a lot more difficult to predict than arousal. Indeed the experiment shows that is is possible to find

correlation coefficient up to 0.35 for valence (weak correlation) while correlation coefficients could go beyond 0.7 for arousal.

Chapter 5

Transfer Learning for Speech Synthesis

Contents

This chapter is based on the following publication:

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Exploring Transfer Learning for Low Resource Emotional TTS". in: *Intelligent Systems and Applications*. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 52–60. ISBN: 978-3-030-29516-5

This chapter is part of the third of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., build a system able to synthesize expressive speech based on the data collected, first based on specific styles.

In this chapter, our goal is to tackle the inherent problem of the amount of data needed in deep learning in the case of emotional TTS which is a topic of growing interest. We can cite [89] that experimented an unsupervised learning technique to change prosody of synthesized sentences with style tokens or [59] that modified Tacotron's architecture to synthesize speech given emotional labels. In this chapter, we explain how to leverage fine-tuning on deep learning-based TTS systems to synthesize emotional speech with a small emotional speech dataset.

Chapter 6 constitutes a practical application of the methodology described in this chapter.

5.1 Introduction

The current state of the art of TTS synthesis is based on deep learning algorithms. These systems are now capable of producing natural human-like speech. However, to reach such results, these systems require a large amount of training data.

Moreover it is difficult to have a fine control on speech quality and emotional content with such systems, while this has become an important challenge in speech synthesis. Here again, data availability is an issue. Indeed, high quality speech datasets with emotional content needed for speech synthesis are quite difficult to collect. The amount of data available is therefore relatively limited compared to what deep learning algorithms require to converge.

Promising methods to tackle the problem of quantity of data are those related to knowledge transfer such as transfer learning [74], fine-tuning and multi-task learning (see Section 2.4.3). These techniques have proved useful in various

applications of deep learning.

In the field of Motion Capture and Analysis, [56] mapped a motion sequence to an Red Green Blue (RGB) image to be able to use a CNN pre-trained for image classification in their motion classification task. The authors showed that fine-tuning the CNN on their motion data improved classification results.

In Chapter 4, we used an neural ASR as a feature extractor for emotion recognition. We showed that the mapping between speech and text learned by the ASR system contains information useful for emotion recognition.

In the TTS field as well, transfer learning is being investigated. In [45], the authors successfully transferred knowledge from a model trained to discriminate between speakers to a multi-speaker TTS model. These examples motivates our interest to investigate the use of knowledge transferability between models.

5.2 System

Our goal is to study the feasibility of fine-tuning a TTS system pre-trained on a big dataset on few new data and analyze how much the model is able to fit them. this section describes the whole system. Figure 5.1 represents its overall idea. First, in Section 5.2.1, we present the TTS system used as a basis for fine-tuning. We then briefly present the dataset we are using in Section 5.2.2. In Section 5.2.3, we explain the pre-processing of our dataset. Finally, in Section 5.2.4, we detail the fine-tuning procedure applied to obtain emotional TTS models.

5.2.1 Text-to-Speech System

The number of deep learning-based TTS system of the state of the art are growing. To carry out our experiments, we chose, DCTTS [93], a system that combines advantages of several systems. DCTTS models a sequence-to-sequence problem with a encoder-decoder structure along with an Attention

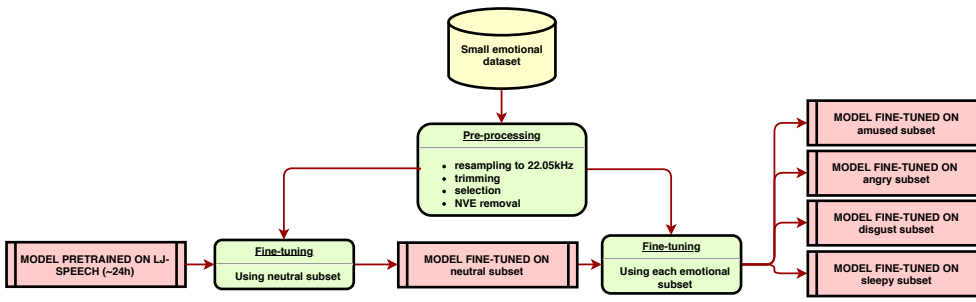

Figure 5.1. Block diagram of the Transfer Learning procedure. A Deep Learning-based system is first trained on a big speech dataset of good quality. A smaller dataset of another speaker that is however richer in emotional expressiveness is pre-processed. The part labeled as neutral is used to fine-tune the pre-trained model to obtain a TTS model fitting the new speaker's voice. Then emotion adaptation is performed by fine-tuning this model with each emotional subset to obtained one model per emotion.

Mechanism like Tacotron [111]. However, unlike Tacotron, the modules of the architecture are all CNN-based and there is no RNN component. In [93], the authors compared an open source implementation of Tacotron to DCTTS and have higher Mean Opinion Score (MOS). In this work, we use the Tensorflow implementation provided in [55].

There are two modules trained separately: Text2Mel and Spectrogram Super-resolution Network (SSRN). Text2Mel takes care of the mapping between character embeddings and the output of Mel Filter Banks (MFBs) applied on the audio signal, i.e., a mel-spectrogram. Text2Mel module models the sequence-to-sequence task. It is composed of a Text Encoder, an Audio Encoder, an Attention Mechanism, and an Audio Decoder. Then the second module SSRN maps the mel-spectrogram to full resolution spectrogram. Finally, Griffin-Lim is used as a vocoder.

5.2.2 Dataset Used

The dataset used in this work is EmoV-DB: The Database of Emotional Voices presented in Chapter 3. To summarize, EmoV-DB contains sentences uttered by male and female actors in English and a male actor in French. Each actor was asked to utter a subset of the sentences from the CMU-Arctic [51] database for English speakers with 5 emotion classes.

In this work we used one of the English actress to perform emotion adaptation of the TTS system. The experiments performed on this dataset also assess its usability with deep learning algorithms for voice generation systems.

5.2.3 Pre-processing

Important aspects of the pre-processing to use this model are:

- the sample frequency;
- the trimming of silences at the beginning and end of audio files;
- removal of nonverbal expressions (laughters, yawns, etc.).

As the model was trained with LJ-speech database with a sample frequency of 22050Hz, we should use the same with our database.

The trimming of silences is important because the model uses guided attention [93]. It helps the attention mechanism by assuming that the ordering of text characters is almost linearly related to the time in the audio file. This is true only if the speech begins from the start of the file without a silence.

We experimented that without this trimming, the synthesized sentences often omitted the first words of the text to pronounce. The implementation of [55] already uses trimming with librosa, a standard audio processing python library [65]. However we noticed that default function parameters were not suited for our database. We thus changed the threshold below reference to consider as silence to $20dB$.

The same problem happens when there are non verbal expressions in the audio files such as laughters, yawns or sighs. Indeed in these cases, the hypothesis to use guided attention is not verified as well. To overcome this problem, we first manually selected utterances without such Non Verbal Expression (NVE) for the amused dataset (156 utterances) and the sleepy dataset (361 utterances). Then for the amused dataset only, we augmented this selection by manually removing laughters from a part of the remaining utterances (82 utterances) to have a total of 238 utterances because the selection was quite small.

5.2.4 Fine-tuning

In this section, we explain how we leveraged knowledge transfer on TTS by fine-tuning a part of a pre-trained model on our small dataset. The pre-training of the model was done using the LJ-Speech dataset. This dataset is available online[1] and contains 23.9h of speech uttered by a single female speaker. The fine-tuning was done with the dataset described in Section 5.2.2.

There are several possibilities of how we could fine-tune the model. First, we can choose which parts of the pre-trained model we want to fine-tune with the new dataset and which part we want to keep fixed.

The second part of the model, SSRN, does the mapping between mel-spectrogram and full spectrogram. Therefore, it should not depend on the speaker identity on speaking style as it is just trained to do the mapping between two audio features. However, as the model has been pre-trained on one speaker, there is a possibility of over-fitting on the characteristics of that specific speaker. The first question we want to answer is whether the SSRN can generalize the mapping to other speaking styles.

As for Text2Mel module, it is composed of a Text Encoder, Audio Encoder, Attention, and Audio Decoder. As the text does not depend on characteristics of the speaker or his speaking style, we tried to train only the Audio part. However we found that there are some problems of rhythm in synthesized

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

speech. We believe this is because Attention module is not adapted to the new speaking style. As a consequence, we chose to fine-tune on the entire Text2Mel module.

5.3 Experiment

In this section, we detail the two experiments performed on the system.

In the first experiment, we evaluate the usefulness of the fine-tuning technique compared to a random initialization of the model parameters. The evaluation is based on a measure of intelligibility of the synthesized speech in terms of word accuracy proposed in [73].

In the second experiment, we evaluate the quality of the emotional speech synthesized through a MOS test for each emotion according to two criteria: confidence in the perception of the emotion specified and the intelligibility.

The amount of speech data used for the experiments are showed in Table 5.1. Durations are rounded to the minute. The values between parentheses correspond to the amount of data before selection and NVE removal.

Table 5.1. Amount of data available for each emotion in terms of total duration and number of utterances.

	Total duration [min]	Number of utterances
Amused	15 (20)	238 (296)
Angry	19	304
Disgusted	29	303
Neutral	23	357
Sleepy	36 (51)	361 (496)

5.3.1 Objective measures

Table 5.2. Intelligibility in terms of Word Accuracy.

	Word Accuracy
LJ-speech	0.630 ± 0.042
Neutral (random initialization)	0.004 ± 0.004
Neutral (fine-tuning)	0.517 ± 0.048

In this experiment, we synthesized 100 sentences of the Harvard sentences [85] with several models. Then an objective measure of the intelligibility of every sentence was computed in terms of word accuracy [73]. The measure consists of using an ASR to recognize speech and compute a word accuracy by comparing the result to the text label.

The mean word accuracy with 95% confidence interval for all models are summarized in Table 5.2. The first line show the word accuracy of the pre-trained model (on LJ-speech dataset). The second line corresponds to the model trained only on the neutral subset. Finally, the third line corresponds to the pre-trained model fine-tuned on the neutral subset.

These measures allow us to compare the fine-tuning of a pre-trained model to random initialization of the model parameters.

The experiments clearly shows that the model trained with the neutral subset of 20 min is unable to generate intelligible speech if the model parameters are randomly initialized. However, if the initialization of the parameters comes from a model previously trained on another large speech dataset, the fine-tuned model leads to a slightly lower intelligibility compared to the reference.

5.3.2 Perception tests

After fine-tuning from the neutral model to emotional models, we synthesized 5 sentences not seen during training with each of these models. These sentences were used in a MOS test. During this test, the participants were asked to complete a form.

This survey contained 5 sections. Each section was dedicated to one emotion. In every section, the participants were asked to rate utterances between 0 and 5 for two criteria:

- the confidence in the perception of the emotion specified (0=we can not hear the emotion specified, 5=we perfectly hear the emotion specified);
- the intelligibility of speech (0=it is indecipherable, 5=it is perfectly intelligible).

This test was performed on both original files from the dataset and synthesized files. Table 5.3 gives MOS with 95% confidence interval for the original files and Table 5.4 gives them for the synthesized files.

Table 5.3. MOS test results of original files.

	Intelligibility	Confidence
Amused	4.47 ± 0.21	4.60 ± 0.20
Angry	4.73 ± 0.18	4.22 ± 0.25
Disgusted	4.42 ± 0.21	3.28 ± 0.27
Neutral	4.83 ± 0.16	4.37 ± 0.23
Sleepy	4.33 ± 0.21	3.80 ± 0.27

Results of Table 5.3 should be considered as higher bounds as they represent the opinion about original files of the dataset. Results from Table 5.4 should be compared to these higher bounds. One can see that these higher bounds do not meet the maximum value of 5.

Table 5.4. MOS test results of synthesized files.

	Intelligibility	Confidence
Amused	2.01 ± 0.24	2.00 ± 0.27
Angry	2.76 ± 0.25	2.10 ± 0.28
Disgusted	2.17 ± 0.27	2.27 ± 0.30
Neutral	3.60 ± 0.26	3.59 ± 0.24
Sleepy	2.59 ± 0.28	3.29 ± 0.26

In Table 5.4, for both intelligibility and confidence of the perception of an emotion, the Neutral category has the higher value. A possible explanation of this is that the pre-trained model used has been trained with a neutral corpus and is therefore closer to the Neutral subset used for fine-tuning. For the other emotional categories, the values are more degraded compared to original samples. These degradations vary depending on the category. The nonverbal components contained in the different categories are different and may be more or less difficult for the system to reproduce depending on the data it has been trained on. For example, the amused category contains chuckled vowels that were not well reproduced. This may explain why it has the greatest degradation. On the contrary, the sleepy category was slow pace and relaxed voice (low tension). These characteristics could be easier to interpolate from neutral speech.

5.4 Conclusions

In this chapter, we present a technique allowing the synthesis of emotional speech using a small emotional speech dataset. This technique is based on the fine-tuning of a deep learning-based TTS model with the neutral subset of the small dataset and then fine-tuning the resulting model with each emotional subset to obtain one model per emotional category.

In the first experiment, we show that training the model with random initialization of the parameters gives completely unintelligible speech synthesis.

However using a pre-trained model as initialization and fine-tuning it leads to more intelligible speech synthesis.

In the second experiment, we perform perception tests to quantify the difference between original recordings from the dataset and synthesized speech for the different emotion categories and with two criteria: the perceived intelligibility and the confidence of detecting the emotion. We believe the variations between MOS score degradations depend on the different speech components and vocal characteristics necessary to generate the different emotion categories.

In applications for which it is needed to keep the speaker identity in the synthesized voice, it would be necessary to evaluate to which extent the vocal characteristics are conserved after the fine-tuning procedure. Indeed, such procedure can lead to an altered speaker identity. This phenomenon is referred to as *speaker leakage*. In this chapter, we do not tackle this aspect as we are mainly interested in synthesizing expressive speech and not controlling or keeping a specific speaker identity.

Chapter 6

Application to Audio Laughter Synthesis

Contents

This chapter is based on the following publication:

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Laughter Synthesis: Combining Seq2seq Modeling with Transfer Learning". In: *Proc. Interspeech 2020*. 2020, pp. 3401–3405. DOI: 10.21437/Interspeech.2020-1423. URL: http://dx.doi.org/10.21437/Interspeech.2020-1423

This chapter is part of the third of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., build a system able to synthesize expressive speech based on the data collected, first based on specific styles. This chapter is an application of the methodology developed in previous chapter.

Despite the growing interest for expressive speech synthesis, synthesis of nonverbal expressions is an under-explored area. In this chapter we propose a system of acoustic laughter synthesis based on a sequence-to-sequence TTS synthesis system. We leverage transfer learning by training a deep learning model to learn to generate both speech and laughs from annotations.

We evaluate our model with a listening test to compare its performance to HMM-based laughter synthesis and assess that it reaches higher naturalness. Our solution is a first step towards a TTS system that would be able to synthesize speech with a control on amusement level with laughter integration.

6.1 Introduction and Motivations

Given the progress in speech technologies and Human-Computer Interactions, several applications of voice assistants and virtual agents have been developed. These applications are evolving towards breaking the barriers between robot-sounding synthetic sounds to human-like conversations. One of the under-explored domains is the synthesis of nonverbal-conversational expressions, particularly laughter. Laughter is an important component of speech and daily interactions. It has been shown to be very frequent cross-cultural expression in conversations, to communicate emotions and to have conversational and social functionalities [27].

Although laughter synthesis systems has already been investigated [107, 67, 27], deep learning-based systems are under-explored for audio laughter synthesis. Indeed, to the best of our knowledge very few work can be found on the subject.

Laughter can be expressed in many different ways and is particular to each individual. It is therefore rather difficult to collect naturalistic genuine laughter in a sound clean environment. This is the main reason why resources available for synthesis purposes are rather limited compared to speech.

In this chapter we present a deep learning-based laughter synthesis system. This work is part of a larger project aiming at synthesizing laughter alongside speech in Human-Agent Interaction systems. In order to build this system considering the limited amount of data available, we leverage the knowledge learned by TTS systems. Indeed, although different, both speech and laughs share common sound characteristics since laughs are sequences of fricatives, vowel-sounds and breathing. We follow a similar approach as in Chapter 5 that used transfer learning for emotional speech synthesis with few data.

The system presented here, improves on previous state of the art in quality and in flexibility. The results of this work is also a stepping stone towards the use of laughter alongside speech and intermingled with it in order to generate speech-laughs [27].

The chapter is organized as follows: related work is summarized in Section 6.2; Section 6.3 presents the datasets involved in this work; Section 6.4 describes the proposed system for audio laughter synthesis; the procedure of the perceptive evaluation is described in Section 6.5; the results of the evaluation are presented and discussed in Section 6.6; finally Section 6.7 gives future works and Section 6.8 draw the conclusions.

6.2 Related Work

The techniques studied for laughter synthesis have generally followed those used for speech synthesis. In this work we will do the same for two main reasons: first, the signals share a lot of common characteristics; second, using TTS systems will allow the incorporation of our laughter synthesis system into a fully functioning TTS system and thus achieve our ultimate goal of TTS

with control over amusement levels.

As detailed in Chapter 2, speech synthesis methods can be grouped in three main categories: synthesis by concatenation, speech production modeling and statistical parametric synthesis.Among the few studies on laughter synthesis, the first attempts included synthesis by diphone concatenation [57], parametric synthesis and by using a mass-spring approach [92]. Then HMM-based models were introduced to laughter synthesis due to their wide-use in speech synthesis back then [107].

In the HMM-based approach of [27], the authors leveraged adaptation of acoustic modeling to transfer knowledge of speech acoustics for the prediction of acoustic features of laughter. Similar domain adaptation techniques have also been studied for emotional speech synthesis with seq2seq deep learning models. In Chapter 5, we tackled the problem of data scarcity of emotional speech by using fine-tuning on a seq2seq TTS model. A model trained to synthesize neutral speech with a large dataset (~ 24 hours) was fine-tuned with smaller speech datasets ($\sim 20 - 40$ minutes) for different categories of emotion.

However, deep learning-based laughter synthesis has been very little explored yet. An approach of synthesis with wavenet was recently proposed by [67]. Wavenet [71] is an autoregressive CNN synthesizing audio sample by sample from features, typically linguistic features for TTS, or acoustic features for vocoding. In their attempt of application to laughter synthesis, the authors conditioned the wavenet model on information of inhalation/exhalation sequence and durations and power contour features predicted by an HMM model. This approach is therefore still relying on previous HMM approaches for a part of the information.

Given the breakthrough of sequence-to-sequence (seq2seq) approaches in speech synthesis systems, we propose an adapted approach for audio laughter synthesis. Also, the method proposed in this chapter offers control over the specific sound sequences to be generated rather than syllable-level control. This offers the flexibility of choosing the specific sequence of voiced (vowel-sounds) and unvoiced to be generated. Given the aforementioned goal of obtaining a fully

functioning TTS system generating laughter alongside speech, another advantage of our system is not only to synthesize naturalistic human-like laughs, but also to do it in a speech context. This will allow a later integration in a fully functioning speech and laugh synthesis system as planned.

6.3 Dataset

In order to apply the transfer learning approach described above, the data used are formed of subsets of a proprietary dataset recorded by Acapela and of the AmuS dataset [30] were taken.

Acapela's dataset was recorded to build a narrating framework to construct book recordings from transcriptions. It contains phonetically rich sentences uttered by a male actor in US English. The actor was asked to utter a set of the sentences in 8 style classes. For the purpose of this work, only the audio recordings of the neutral style were kept along of the corresponding transcription with a total of 150.50 minutes (3299 utterances) of speech data.

The AmuS dataset contains recordings of amused speech components such as smiled speech, laughs and speech laughs. For this work, the laughs coming from the speaker with the most laughs (SpkB).

In this dataset, the purpose of these laughs were to be inserted in speech in order to create amused speech which suites well with the goal of generating laughs alongside speech as mentioned above. So in order to record these, the subject was asked to watch funny stimuli while sustaining a vowel, until eventually laughter occurred naturally interrupting the vowel. This would allow us to collect laughs with transitions from vowels. We thus have at our disposal laughter occurring in three vowel contexts: [a], [e], [i] (French International Phonetic Alphabet (IPA) symbols). Table 6.1 breaks down amount of data available.

Vowel	[a]	[e]	[i]	Total
number of samples	54	33	25	112
duration (sec)	101	63	38	202

Table 6.1. Quantity of laughs per vowel context.

Isolated laughs are sequences of voiced and unvoiced sounds. In AmuS, the laughs were segmented and each segment given a label corresponding to a voiced or unvoiced categories.

It is these label sequences and the corresponding laughter audio signals that were used to train our systems.

The annotation of laughter in AmuS dataset is as follows:

- '[a,e,i]Block': Voiced part of laugh - ["a", "e", or "i"] sound occuring but blocked or cut to utter the following sound;
- '[a,e,i]L': Voiced part of laugh - ["a", "e", or "i"] sound;
- '[a,e,i]N': Sustained vowel ["a", "e", or "i"];
- '[a,e,i]Up': Voiced part of laugh - ["a", "e", or "i"] sound, following "[a,e,i]N" with increasing pitch;
- 'highPitched': Voiced part of laughter - sound not very distinguishable with high pitch;
- 'inh(L)': inhalation (following laugh);
- 'UnvoicedChuckles': Unvoiced chuckling laughter;
- 'pulse(In)': Unvoiced part of laughter (at the beginning);
- 'short': Voiced part of laughter - sound not very distinguishable;
- 'sil': No sound;
- 'silentLaugh': laughter occuring but silent.

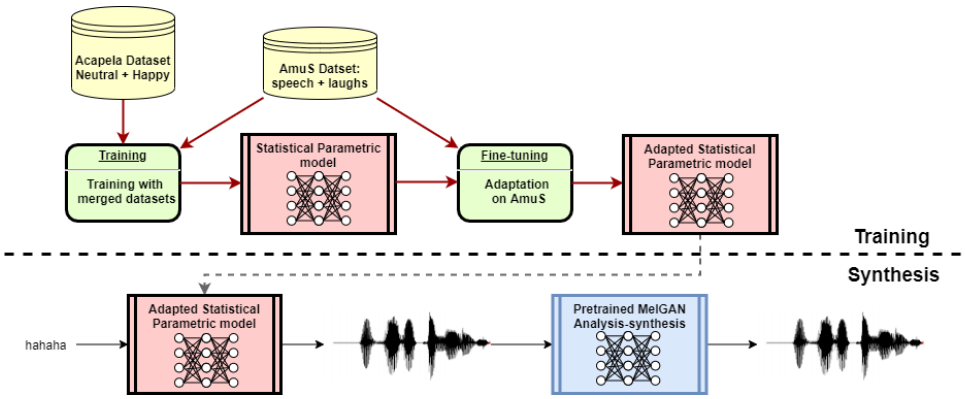

Figure 6.1. Block diagram of the proposed method for model adaptation.

6.4 Seq2seq Audio Laughter synthesis

Figure 6.1 shows a block diagram of the procedure proposed for model adaptation and waveform correction with MelGAN.

6.4.1 System description

Nowadays, one of the major techniques for Text-to-Speech synthesis are deep learning architectures based on the sequence-to-sequence (seq2seq) principle. It consists of an encoder-decoder setup with an interface between the two components called *Attention Mechanism* whose role is to model the alignment between input and output sequences. Well known seq2seq TTS systems are Tacotron [111], Char2wav [91] and DCTTS [93]. In this work we adapted DCTTS model for audio laughter synthesis using an open implementation available online[1].

In this work, the input sequence is composed of speech phonemes and laughter annotations described in Section 6.3. We use festival [7] to extract cmu phones from transcriptions in the Acapela dataset and the speech part of AmuS dataset. The output sequence is a mel-spectrogram. A second part of

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

DCTTS, trained separately reconstruct a full resolution magnitude spectrogram from the mel-spectrogram to be inverted to a waveform using Griffin-Lim algorithm [38]. For more details about these, see [93].

The DCTTS system is first trained with the two merged datasets. Then domain adaptation is performed by fine-tuning the model with the subset of the data containing the voice of AmuS with both speech and laughters.

6.4.2 Waveform correction with MelGAN

To generate the waveform from acoustic features, it has been shown that neural audio synthesizers achieve better quality in terms of naturalness [71, 46]. However it is generally a challenge to design and optimize such models efficiently to reach the expected results described in the literature. They also often loose generalization properties compared to signal processing based vocoders, as they are often speaker dependent.

MelGAN [54] is a recently proposed model that tackled the problems of efficiency and generalization across speakers. The model is non-autoregressive, fully convolutional and smaller then previous ones.

In this chapter we use MelGAN as a waveform corrector. The laughter waveform is first synthesized by the system described in Section 6.4.1. This waveform contains artifacts due to the Griffin-Lim estimation. Then we apply analysis and synthesis with MelGAN to obtain a corrected waveform that we show is better in terms of perceived naturalness.

6.5 Evaluation

6.5.1 Perception Tests

For the perception test, we gathered the samples from these different methods:

- Method 1: laughter samples from AmuS dataset unmodified;

- Method 2: synthesis based on the HMM-based speech synthesis system (HTS) equivalent to the one used in [27];
- Method 3: seq2seq model (seq2seq-GL) trained on Will speech dataset and AmuS speech and laughter data with adaptation as described in Section 6.4.1;
- Method 4: same seq2seq model as method 3 along with the proposed MelGAN waveform correction (seq2seq-MelGAN).

Method 1 is there to have a reference as real laughter do not achieve perfect scores. It gives therefore a topline of naturalness for the different method. A listening test (MOS test) was performed to evaluate the naturalness of the synthesized samples of the methods. The MOS test was implemented as a web experiment with turkle[2], which is an open-source web server equivalent to Amazon's Mechanical Turk that one can host on a server or run on a local computer.

The samples were presented to the listener in a random order one by one. He could listen to each sample as much times as needed. The listener was asked to rate each samples in terms of naturalness a 5-point Likert scale with the following labels:

- 5 - very natural;
- 4 - natural;
- 3 - fairly natural;
- 2 - unnatural;
- 1 - very unnantural.

The definition of naturalness for speech and laughter could be interpreted differently during evaluation. When rating laughter, the listeners could rate the acting quality instead of human-likeness. As a component of not natural can be "fake" or "simulated" laughter instead of synthetic. We thus added an explanation of the meaning of natural after the question: "How natural does the laugh sound ? By natural, we mean that the audio sounds human-like."

[2] https://github.com/hltcoe/turkle

	Female	Male	Sum
[20,40[	7	13	20
[50,65[	1	3	4
Sum	8	16	24

Table 6.2. Number of participants by gender and age range (in years).

	#ratings	MOS	std
HMM	407	2.64	1.02
seq2seq-GL	431	2.50	1.09
seq2seq-melgan	429	3.28	1.06
original	429	4.10	0.91

Table 6.3. Number of collected ratings, MOS scores and their standard deviation for each method.

Twenty four listeners completed the perception tests. They were composed of 16 males and 8 females (see Table 6.2).

6.6 Results

6.6.1 Quantitative Analysis

A total of 1696 answers were collected. Table 6.3 gathers the number of ratings for each method, the resulting MOS scores and their standard deviation.

Figure 6.2 shows the distributions of the ratings as boxplots. Figure 6.3 shows the percentage of scores chosen for each method. Original samples of the dataset reach a MOS score that is below 5 as listeners do not rate all original samples as perfect. There is a quite high variance, that is close to one for all

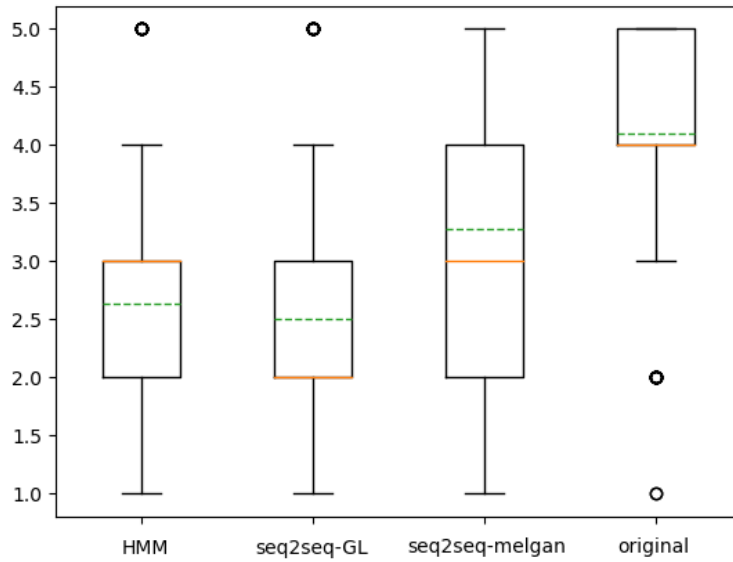

Figure 6.2. Boxplots of scores distributions of the different methods. The green lines correspond to the Mean Opinion Scores.

methods, which was also the case in [107].

Although MOS tests are never executed exactly the same way, it is interesting to analyse the contribution of the different blocks in the degradation of the MOS scores of the experiments of this work with [107] and [54]. In [107], the authors compared different variants of HMM-based laughter synthesis. In [54], a comparison of MelGAN vocoder to Griffin-Lim algorithm for speech synthesis is provided.

In [54], for seq2seq TTS, Griffin-Lim is responsible of a large part of the distortion leading to a loss of 2.95 MOS points compared to original samples. MelGAN offered a gain of 1.77 points of MOS over Griffin-Lim algorithm. In this work, the MelGAN waveform correction offered a gain of 0.78 points of

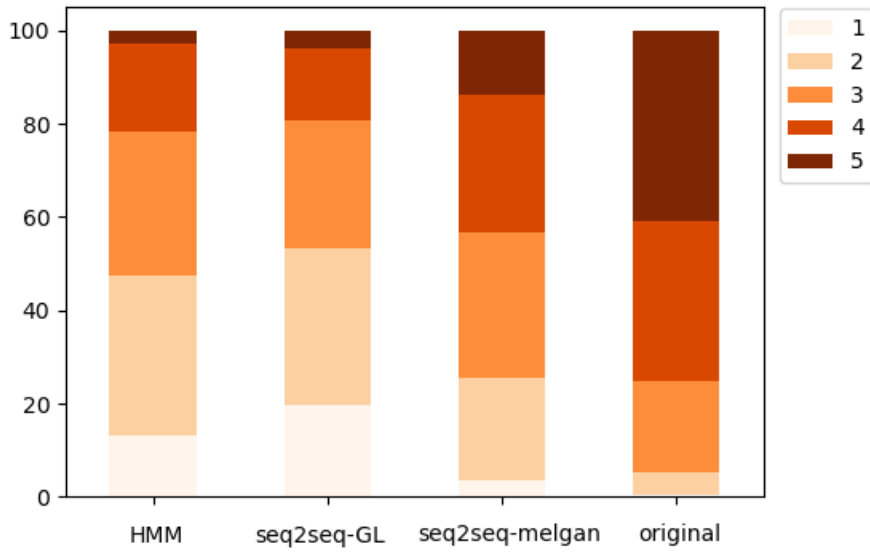

Figure 6.3. Score distributions of the different methods.

MOS. It is important to highlight the fact that training MelGAN directly on generated spectrograms will likely improve these results over this waveform correction by analysis-resynthesis with a pre-trained MelGAN model.

In [107], the distortion caused by the vocoder is of 0.8 compared to original samples. Their best synthesis solution is 0.6 points below that, and therefore 1.4 below original samples. In this work, the HMM approach is 1.46 below original samples which is close to their results. In [8], the authors confirm that HMM-based laughter synthesis have significantly lower quality than copy-synthesis with several vocoders. The seq2seq-melgan method is our best solution and still has potential of improvement.

Figure 6.4 shows the evolution of the MOS score depending on the duration of the samples. In all techniques, we can observe a trend of an increasing

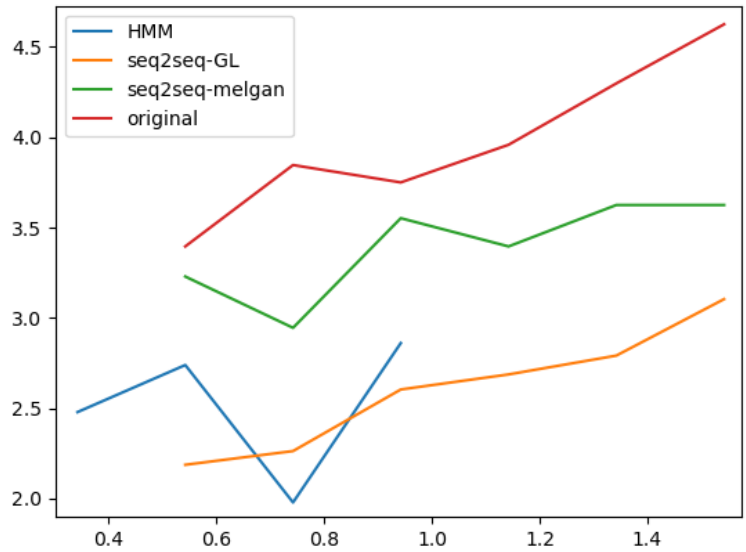

Figure 6.4. MOS Scores evolution depending on sample durations by taking intervals of 0.2.

score when the laughters last longer. This phenomenon was commented by the participants of the listening test as explained in the section hereafter.

6.6.2 Qualitative Analysis

A part of the participants were asked to explain the different characteristics that, according to them, made them rate high or low. Some important components of laughter were specific noises that people make when laughing and were perceived as especially natural: the inhalation at the end of the laughs, little coughs or noise to clear its voice, noise of someone trying to breath.

A monotonous prosody decreases the naturalness. Having several variations in pitch and duration in a laugh was perceived as more natural. Laughs sounding

like a repeated sequence were perceived as robotic whereas laughs containing more randomness were perceived more natural.

Also the duration of laughter seemed important. Some participants reported that short laughter were harder to evaluate as they thought it could be real laughter but very short to be able to do the distinction. For others, the naturalness is increased when the duration increase as very short laughters seem unlikely. One participant stated however that some laughters seemed too long to be natural. A progressive diminution of laughter loudness was perceived as more natural than an abrupt end of laughter.

Different noises generated in the samples were perceived as annoying and decreased the perceived naturalness. Some participants described them as *something virtual*, *low sound quality* or *robotic*.

The fricatives in HMM based synthesis was perceived as too pronounced white noise. The end of HMM laughs were perceived as too abrupt. It is well known that HMM synthesis sounds buzzy. It was also described as containing "weird vibrations". The noise within seq2seq-GL samples were described as acute trebles and metallic. We believe this is mostly due to Griffin-Lim algorithm that is known to produce metallic-sounding audio.

6.7 Future Works

This system is meant to be integrated in a full TTS system in which the level of amusement in speech can be controlled [24], speech-laugh could be generated and laughter can be synthesized interrupting or accompanying speech. We believe several modifications could improve the acoustic quality of the synthesis. First end-to-end training could help concerning the accumulation of errors of several blocks: the seq2seq system and the vocoder.

6.8 Conclusions

A new approach of Audio Laughter Synthesis based on seq2seq learning was proposed inspired by the evolution of TTS field. We propose to train a deep learning system to synthesize speech and laughter from transcriptions by augmenting the input of phonemes with laughter annotations. This method leverages transfer learning of patterns between annotations and acoustic features of both worlds.

We show that using a pre-trained MelGAN model as a post waveform corrector allows the removal of audio artifacts generated by Griffin-Lim algorithm.

The goal of this Chapter was not to give a thorough comparison of HMM and DNN methods, as audio quality of these are already well studied in the state of the art [114, 113], but rather propose a method that is better than current state of the art and in line with current speech technologies. This results in a strong improvement over past methods of audio laughter synthesis in terms of naturalness and is promising for synthesizing speech-laughs thanks to a consistency with latest speech synthesis technologies using seq2seq approaches.

Chapter 7

Perceptual Analysis of Controllable Speech Synthesis

Contents

This chapter is based on the following publication:

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Neural Speech Synthesis with Style Intensity Interpolation: A Perceptual Analysis". In: *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '20. Cambridge, United Kingdom: Association for Computing Machinery, 2020, pp. 485–487. ISBN: 9781450370578. DOI: 10.1145/3371382.3378297. URL: https://doi.org/10.1145/3371382.3378297

This chapter is part of the last of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., controlling an expressive speech synthesis system with a representation of emotional expressiveness.

State of the art in speech synthesis allows generation of almost human sounding speech and also allows some control on the style. However the impact of such control on the perception between styles is not well known. Are the control variables useful and do they control what we expect ?

In this chapter, we propose a method to evaluate the impact of control variables on the perception of the speech style through a perceptual analysis. We train a speech synthesis system with different discrete style categories and study its ability to interpolate continuously between neutral speech and styled speech with an intensity control variable. We then measure the effect of this interpolation of control variables on the perception of speech through listening tests.

The results show that despite being trained with discrete categories encoded as one hot vectors, it is possible to synthesize utterances that are perceived as intermediate levels of intensities between neutral and styled speech.

7.1 Introduction

In Human Agent Interaction (HAI), one of the main task is Text-to-Speech synthesis allowing the agent to communicate information to the user. In recent years, many TTS systems have been successfully developed through deep learning algorithms, leading to a synthesis very close to natural human speech. This problem being solved, the interest towards controlling the style of synthesized speech have grown [112, 90].

Some systems able to build latent representations of styles [45, 90] have been successfully developed. These systems are able to interpolate in the space of this representation. For this purpose, these systems need to be trained on a large amount of very diverse data in terms of speakers and styles that is

expensive to collect.

In this chapter, we study the capabilities of a Deep Learning-based Speech Synthesis system to synthesize different intensities of style while being trained with a reduced dataset made of six distinct style categories.

The effects of such control on the perception of the style by humans are not well known. The typical test to evaluate the synthesis subjectively being a MOS test that only takes the naturalness into account.

In this chapter, we aim to study the perception of style interpolation of our controllable TTS system. The chosen metric is based on comparison of pairs of synthesized utterances.

7.2 Dataset

The dataset used in this work was recorded by Acapela Group SA. It consists of recordings of a male english-speaking actor that was asked to read predefined sentences with different styles: neutral, happy, sad, bad guy, from afar, whispering, old man. For every style, recordings contain approximately 2 hours of speech. For more details, see the description in Chapter 3.

7.3 Model

The system consists of a multi-style TTS system able to control the intensity of a given style category. It is a modified version of DCTTS [93], a Deep Learning-based TTS system. In the modified version, it takes an encoding of the category at the input of the encoder. During training, a simple one-hot encoding is used, i.e., a code of 7 dimensions corresponding to the different styles. A 0 is assigned to all dimensions except for the style for which a 1 is assigned.

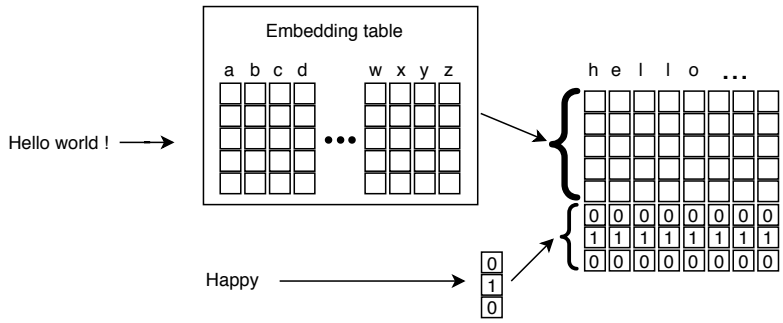

Figure 7.1. Input for multi-style TTS : the characters are encoded in a matrix of N columns of size e via an embedding table like in the original version. In this version, the style is encoded as a one-hot vector and *broadcast-concatenated* to the encoded input text. In this Figure, $e = 5$ and the number of styles would be 3. In the real system, $e = 128$ and the number of styles is 7.

The procedure to obtain the input is represented in Figure 7.1. The text is represented with a sequence of N characters, each one embedded in a column vector of size e. Therefore, it is a matrix of shape (e, N). The 7-d style vector is repeated N times to have a matrix of dimensions $(7, N)$. These two matrices are concatenated to have a matrix of shape $(e + 7, N)$. And this matrix is fed to the rest of the architecture. In other words, the style vector is *broadcast-concatenated* to the transcription embedding. At synthesis stage, we can modify the intensity of a style category by interpolation between codes.

7.4 Analysis of the Impact of Control Variables on Style Perception

We want to know whether the model is able to interpolate intensities without having seen intermediate styles in the database. In this experiment, we interpolate between neutral and each style. Only the number corresponding to neutral and the category with which we interpolate are non-zero. The sum of the numbers of a code is one.

A fictive example of code to have *fairly angry* speech with three categories (neutral, happy, angry) would be [0.3, 0, 0.7]. We will call control variable the number corresponding to the non-neutral category, i.e., 0.7 in this example.

To study the relationship between the control variables and the perception of styles in synthesized utterances, listening tests have been performed. The subjective task for a human of assigning an absolute value of intensity to a subjective concept is not very accurate. Humans are on the other hand better to compare and classify elements. This is related to the ordinal nature of emotion. In [117] the authors advise to design tests with a comparison of pairs of elements. In Speech Evaluation, the first method corresponds to MOS listening test and the second one corresponds to CMOS listening test (or AB test).

CMOS test allows the comparison of two systems by comparing many pairs of files. The result gives a score quantifying to what extent System 1 is better or worst than System 2. However it can not be used to analyze the relationship between a control variable and the subjective evaluation of a criterion such as naturalness or perceived emotion intensity. However comparing lots of pairs of audio files with a given intensity can give a ranking and a score for each intensity.

In this work, we use this kind of score and we analyze its correlation with control variables to observe their effect on the perception of styles.

The database contains a neutral category and six other styles. For each style, we synthesize five different utterances from the standard Harvard sentences [85]. These are synthesized with five different intensity levels: 0, 0.25, 0.5, 0.75 and 1. This makes a total of 150 synthesized files. The number of pairs within a style is the number of combination of two elements among five:

$$^{n}C_{k} = \frac{n!}{k!(n-k)!} = \frac{5!}{2!(3)!} = 10 \tag{7.1}$$

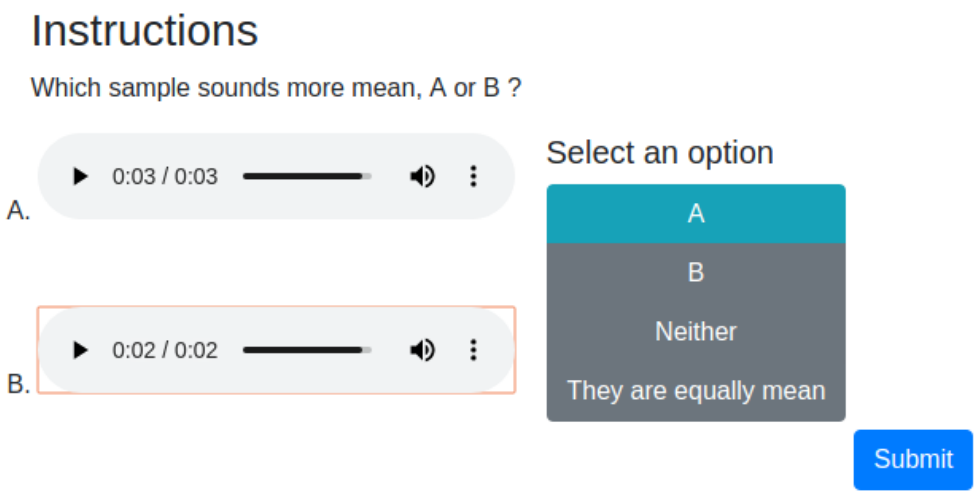

Figure 7.2. Template of question of the listening test.

For 6 styles, 5 sentences and 10 pairs of intensity levels, there is a total of 300 pairs.

The listening test was implemented with the help of turkle[1] mentioned in Chapter 6, an open-source Mechanical Turk platform (similar to Amazon Mechanical Turk) that can be run locally. It leverages the possibility to design a template task to be done on a series of cases, ask participant to solve them, and record results automatically in a database. The template task designed for this listening test is illustrated in Figure 7.2. In each task, the participant has to select between: ['A', 'B', 'Neither', 'They sound equally *style*'].

To ensure the absence of information about style in the system, the filenames were generated randomly, the order of the audio files in a task is randomized, and the order between the pairs shown in tasks is randomized. Each task of the listening test will show a pair of sentence. The question is "Which sample sounds more *style*, A or B ?".

[1] https://github.com/hltcoe/turkle

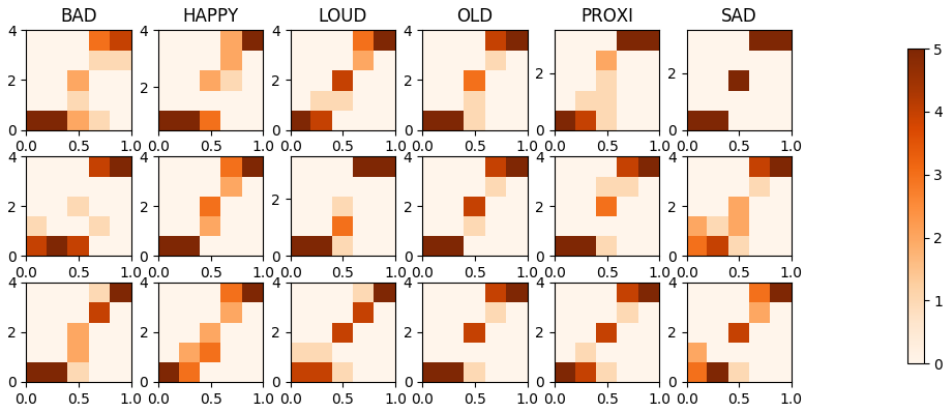

Figure 7.3. 2D Histogram of the number of associations between accumulated scores representing the perception of the intensity level of the style and control variables. From top to bottom, each line correspond to Subject 1 to 3 respectively. The y-axis is the accumulated score from 0 to 4. The x-axis is the control variable. The color correspond to the number of times a participant associated x to y.

Each file will appear in 4 comparisons for each listening test. For each comparison, a score of 1 is assigned to the winner and 0 to the looser. If equal is selected, 0.5 will be attributed to both files. If neither is selected, a score of 0 will be assigned to each file. For each file, the scores are then summed to obtain a score representing the perceived intensity of the style.

In the ideal case in which the control variable would exactly correspond to the perception of the participant, the scores corresponding to the intensities 0, 0.25, 0.5, 0.75 and 1, would be 0, 1, 2, 3 and 4, respectively.

7.5 Results

To visualise the relationship between the perceived intensities and the control variables, we show a 2D histogram of the values of both variables by category and by subject in Figure 7.3. To quantify this relationship, we compute the

Table 7.1. Pearson Correlation Coefficient between control variables and perceived intensities by subject and by category

	bad	happy	loud	old	whisp.	sad	Overall
Subject 1	0.862	0.914	0.953	0.931	0.913	0.947	0.913
Subject 2	0.839	0.942	0.917	0.948	0.924	0.881	0.904
Subject 3	0.944	0.946	0.929	0.929	0.930	0.900	0.929

Pearson Correlation coefficient between these two variables (Table 7.1).

The listening test was done on three participants. An expert in the field familiarized with the content of the database and the synthesis system (Subject 3) and two other people: one female (Subject 1) and one male (Subject 2). It has been proven empirically that there is a relationship between representational variability of a listener and its ability to perceive synthetic speech [2].

The study is done on three participants, which is not much. The fact that the results of the three participants are similar tends to indicate that it does not vary too much. However, the conclusions of this study should be taken with care.

An estimation of the mean and standard deviation of the Pearson Correlation Coefficient between the intensities and a random choice can be computed. We sample 150 elements of a uniform discrete distribution and compute the Pearson correlation coefficient. By repeating this n times with $n = 10000$, we have a distribution with a mean of 0 and standard deviation of 0.081.

This result shows that the perception of style is highly correlated to the control variable of the DNN and that the DNN is able to successfully interpolate between styles even though it was trained on 0 and 1 labels.

7.6 Conclusions and Future Works

This chapter studies the ability of a Deep Learning-based System to synthesize speech and interpolate an intensity of style while being trained on specific categories with one-hot labels. The results show that a listening test based on comparison of pairs of audio files synthesized with different intensities of style lead to a perception score highly correlated to the intensity initially used for the synthesis.

The proposed system is thus able to interpolate between neutral and a specific style, i.e., to control the intensity of this style. However intermediate levels seem difficult to distinguish compared to extreme ones. Indeed more confusion can be observed in these intermediate levels.

Also, trying to interpolate between different styles results in unnatural speech. We believe that this is because the encoding based on one hot encodings is not adapted to model the relationships between the styles.

To overcome this problem, a latent space designed to represent the variability could result in a more general interpolation technique between styles. A similar study could then be performed on synthesized utterances. The use of a VAE could be a possibility to improve the interpolation in intermediate levels. This framework has the property to improve generalization by forcing a continuous distribution of encodings.

Chapter 8

Latent Spaces for Controllable Speech Synthesis

Contents

This chapter is based on the following publication:

- Noé Tits, Fengna Wang, Kevin El Haddad, Vincent Pagel, and Thierry Dutoit. "Visualization and Interpretation of Latent Spaces for Controlling Expressive Speech Synthesis through Audio Analysis". In: *Proc. Interspeech 2019*. 2019, pp. 4475–4479. DOI: 10.21437/Interspeech.

2019-1426. URL: http://dx.doi.org/10.21437/Interspeech.2019-1426

This chapter is part of the last of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., controlling an expressive speech synthesis system with a representation of emotional expressiveness. Inspired by the results of Chapter 4, this information is extracted from Deep Learning architectures.

The field of TTS synthesis has experienced huge improvements last years benefiting from deep learning techniques. Producing realistic speech becomes possible now. As a consequence, the research on the control of the expressiveness, allowing the generation of speech in different styles or manners, has attracted increasing attention lately. Systems able to control speech synthesis have been developed and show impressive results. However the control parameters often consist of latent variables and remain complex to interpret.

In this chapter, we analyze and compare different latent spaces and obtain an interpretation of their influence on expressive speech synthesis. This will enable the possibility to build controllable speech synthesis systems with an understandable behaviour.

8.1 Introduction

During the last few years, many Text-to-Speech systems based on Deep Learning were developed and showed remarkable performance in terms of reliability and quality of speech. Lately, researchers in this area have been focusing on controlling the speech variability of this kind of systems.

An issue for such a control is the lack of data labeled with information such as emotion or style. Emotion modeling is thus one of the main remaining challenges in the aim of developing more natural human-machine interfaces. Two main approaches exist to model emotions.

A first representation is the categorical representation, such as Ekman's six basic emotions model [22] which identifies anger, disgust, fear, happiness, sad-

ness and surprise as six basic emotions from which the other emotions may be derived. Some speech datasets are annotated in emotional categories. This kind of datasets allows the development of category-based emotional TTS as in Chapters 5 and 7 or in [59]. The disadvantage of such a simple annotation is that they do not offer a continuous representation of emotion.

Emotions can also be represented in a multidimensional continuous space like in the Russel's circumplex model [86]. This modeling approach better deals with the complexity and the variations in the expressions, unlike the category system. The two most commonly used dimensions in the literature are the arousal, corresponding to the level of excitation and the valence corresponding to the pleasure level or positiveness of the emotion. In datasets [14, 13] annotated in emotional dimensions, for each utterance, the final emotion value were obtained by averaging over all annotated results from raters.

However they are not suitable for synthesis purpose. Indeed, although the data is well recorded and post-processed, it contains overlapped speech segments due to the data recording setup (dyadic conversation) and some external noise. Moreover, humans are not reliable for giving absolute values to estimate subjective emotional variables [66].

Moreover, in the field of psychology, it is difficult to find a consensus[1] about the best way to portray emotion, e.g., the often used valence-arousal model has not been well verified, isolated and little reproduced. In this Chapter, the approach is thus oriented towards styles and vocal characteristics.

As mentioned in Chapter 4, some researchers tackled the problem of how to capture emotional representation by training systems on other tasks, leading to different approaches employing transfer learning techniques.

To achieve controllable expressive speech synthesis, recent researches have proposed supervised techniques based on prosody features and unsupervised

[1] https://www.theguardian.com/books/2020/sep/25/im-extremely-controversial-the-psychologist-rethinking-human-emotion

techniques avoiding the problem of labels. Different approaches are explained in the following section.

8.2 Related work

A task related to controllable expressive speech synthesis is the task of prosody transfer for which the goal is to synthesize speech from text with a prosody similar to another audio reference. A common characteristic of both tasks is the need for a representation of expressiveness. However, for controllable speech synthesis, this representation should be a good summary of expressiveness information, i.e., they should be interpretable and in a reduced number of dimensions. For prosody transfer, the representation should be as accurate and precise as possible. In this section we give an overview of different systems for both tasks and situate our approach in the field. Some works analyze systems for both tasks while others focus on one of them.

In [90], the authors present a prosody transfer system extending the Tacotron speech synthesis architecture. This extension learns a latent embedding space by encoding audio into a vector that conditions Tacotron along with the text representation. These latent embeddings model the remaining variation in speech signals after accounting for variation due to phonetics, speaker identity, and channel effects.

In [50], the authors propose a supervised approach that uses a time-dependent prosody representation based on F_0 and the first mel generalized cepstral coefficient (representing energy). They use a dedicated attention module and a Variational Auto Encoder (VAE) to be able to concatenate this information to linguistic encodings. This allows for a fine-grained prosody transfer instead of a sentence level prosody information.

CopyCat [47] addresses the problem of speaker leakage in many-to-many prosody transfer. This problem occurs when the voice of the reference sample can be heard in the resulting synthesized speech while it should only transfer prosody and not speaker identity. They are able to reduce the phenomenon with a

novel reference encoder architecture that captures temporal prosodic representations robust to speaker leakage.

Concerning controllable speech synthesis, [1] proposed to use a VAE and deploy a speech synthesis system that combines VAE with VoiceLoop [94]. Some other researches have then used the concept of VAE [43, 40] for controllable speech synthesis. In [43], the authors combine VAE and GMM and call it Gaussian Mixture Variational Autoencoder (GMVAE). For more details concerning the different variants of such methods, an in-depth study of methods for unsupervised learning of control in speech synthesis is given in [40]. These works show that it is possible to build a latent space leading to variables that can be used to control the style of synthesized speech.

In [112], the authors show an example of spectrograms corresponding to a text synthesized with different rhythms, speaking rates and F_0. However these works do not provide insights about the relationships between the computed latent spaces and the controllable audio characteristics.

Different supervised approaches were also proposed to control specific characteristics of expressiveness [88, 83]. In these approaches, it is necessary to make a choice of control parameters, a priori, such as pitch, pitch range, phone duration, energy, and spectral tilt. This reduces the possibilities of the controlability of the speech synthesis system.

In this chapter, we aim to build latent spaces from an audio dataset containing different speech styles. We want the latent spaces to be useful to control a speech synthesis system. We use typical feature selection techniques as a way to compare the different embedding types in terms of style discrimination ability. We then study relationships between each latent space and audio features to investigate the remaining variability that could be used to control speech generation.

For this purpose, we compare three latent spaces computed by training deep learning-based systems on three different tasks:

- Style classification;
- Speaker classification;
- Text-to-Speech with a Style Encoder.

With the use of a learned latent space, it has been shown that it is possible to generate consistent styled speech by varying their values. Thanks to the analysis of this work, we will also know how the latent variables influence the audio features and be able to control them as we want.

8.3 Dataset Used

In this work, we experiment with two of the datasets described in Chapter 3. The dataset used in Section 8.4 and 8.5 is a proprietary dataset of ACAPELA GROUP SA. It is constituted on a set of very different styles.

The dataset used in Section 8.6 comes with a segmentation in sentences with their corresponding audio. But as the speaker often changes its voice style in different breath groups, we segmented it in voice groups. Phonetic transcriptions were first extracted. Then we used forced alignment on the dataset to have the timing of phonemes and pauses. Then the dataset was segmented further thanks to the pauses in breath groups.

8.4 Embedding Computation Systems

In this section, we describe the workflow of embedding computation. The three tasks used to generate embeddings are: Style Classification, Speaker Classification and TTS with style encoding. Figure 8.1 illustrates how the data is used to train the different systems. To make a fair comparison among the three tasks, we restrict the resulting embedding to a 8-dimensional vector for each task

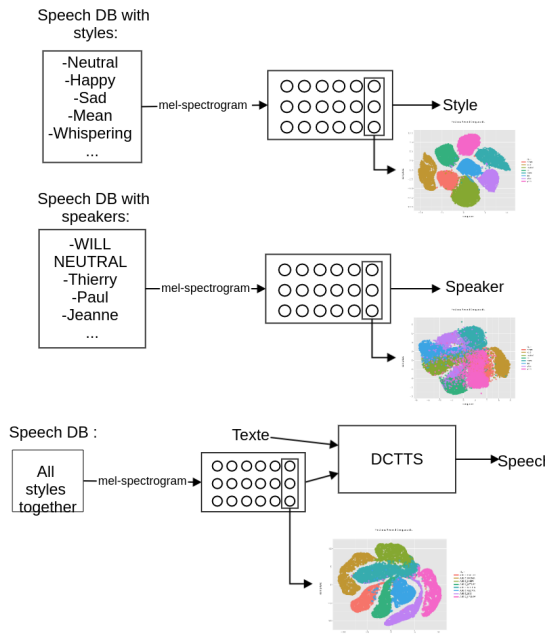

Figure 8.1. Diagram of three embeddings computations systems. From top to bottom: the first system is trained to predict the style label from a mel-spectrogram. The last layer before classification is the latent space and can be visualized in 2D after dimensionality reduction. The second one works in two steps. It is first trained to predict the speaker label of a dataset constitued with many speakers with neutral voice. Then at inference we forward pass mel-spectrograms of other styles. The latent space will show an intra-speaker variance. That is again visualized in 2D. The last system is a combination of an audio encoder trained along with a TTS system. The last layer of the audio encoder is concatenated with text information and passed to the TTS system. The latent space is thus intended to represent variation in the audio that is complementary to text information.

8.4.1 Style Classification System

The classification system is an LSTM-based DNN trained to predict the style category from audio features. The DNN consists of three LSTM layers (512,128

and 64 cells in layers 1,2 and 3, respectively) and a fully connected 8D embedding layer. The input is 80-bin mel-spectrogram of utterances. The generated embeddings capture discriminative information related to the class it belongs to. In our case, the classes refer to the 8 styles of Will voice.

Similar to [45], training utterances are firstly segmented into fixed length segments without silence and the embedding from the last frame is taken as the style embedding of the considered segment.

In experiments, we tried different segment lengths and found out that a length of 800 ms gave the best classification performance. In evaluation stage, we feed the whole utterance to the DNN to get the corresponding style embedding. In contrast with the speaker verification model in [110], our style classifier directly uses style labels to compute loss without calculating the cosine similarity matrix. As our interest is to investigate latent embeddings that help to design an expressive speech synthesis system, the priority is to have embeddings that contain useful emotional or style information, the style classification accuracy becomes then a secondary target. Even tough it was not the main purpose, the accuracy on style classification reached the impressive score of 97% on the 800 unseen utterances (100 utterance per style).

8.4.2 Speaker Classification System

As in Section 8.4.1, the speaker classification system is also an LSTM-based DNN. Here we use four stacking LSTM with the following architecture: 512/512/512/64 cells in layers 1, 2, 3 and 4, respectively and an 8D embedding layer.

In this case, the system is trained to predict the speaker identity from audio features. The speaker classifier is trained with the voice of 276 speakers including the neutral subset of Will voice, the other styles of Will are not used during training.

The dataset is composed of speakers of 32 languages, and different ages (child to elder people), gender and personality as well. We would expect the embeddings generated from such a speaker classifier to reflect information of not only language, age, gender associated, but also prosody associated. Same as the style classification system, utterances are firstly segmented into 800 ms segments without silence in the training stage, and the full utterance is fed to the classifier to get its embedding vector from the last non-silent frame.

8.4.3 TTS System with Unsupervised Style Encoding

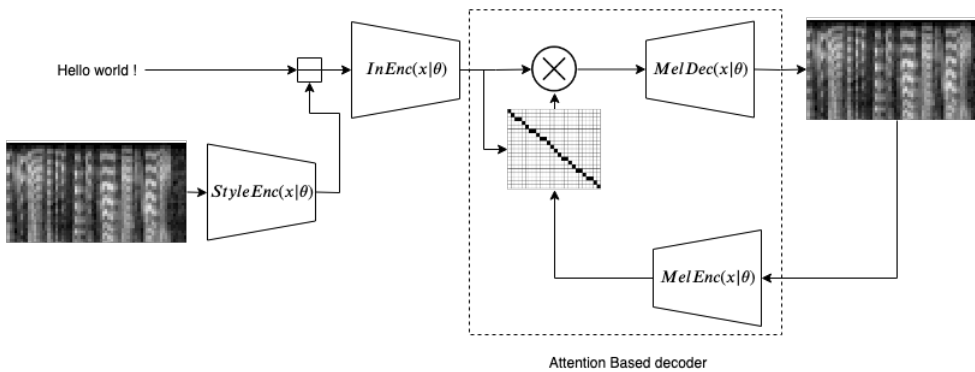

Figure 8.2. Block diagram of the system.

The system is a Deep Learning-based TTS system trained to predict a spectrogram from characters that was modified to enable a control on acoustic features through a latent representation. The basis system is DCTTS [93]. Figure 8.2 shows a diagram of the whole system. The DCTTS system is constituted of the $InEnc$ and the Attention-based decoder that encompasses $MelEnc$ and $MelDec$, as explained in Section 2.11.

For the latent space design, the $StyleEnc$ network was added. It consists of a stack of 1D convolutional layers similar to the $MelEnc$, followed by an average pooling. This operation enforces to encode time independent information. It can thus contain information about statistics of prosody such as pitch aver-

age, average speaking rate, but not pitch evolution. The latent vector at the output is the representation of expressiveness. This vector is then broadcast-concatenated. For the latent vector, we chose a dimension of 8 to be relatively small compared to mel-spectrogram size (80 bins).

There are two modules trained separately: Text2Mel and SSRN. Text2Mel performs the mapping between character embeddings and a mel-spectrogram, then the second module SSRN does the mapping between the mel-spectrogram and full resolution spectrogram. Finally, Griffin-Lim algorithm is used as a vocoder.

8.5 Audio Analysis and Interpretation of Latent Spaces

The latent representation should be linked to the style information. This is assessed in Section 8.5.1. We use typical feature selection techniques as a way to compare the different embedding types to evaluate their ability to discriminate between styles.

Latent spaces computed in Section 8.4 should be useful to control a speech synthesis system. The goal is thus to have a latent space that has interpretable relationships with audio features that can be used to control speech generation.

This is investigated in Section 8.5.2 through an analysis of the correlations between audio features and a linear approximation of these features that uses embeddings. In Section 8.5.3, we present a 2D visualization technique of these relationships.

8.5.1 Style Classification score

Embeddings computed from the style classification system were trained to have a high style classification score. Here we investigate the suitability of the two other embedding types for a style classification task.

We compare the suitability of embedding types for classification using typical feature selection methods that analyses the relationship between these embed-

dings and the categories. The first is ANOVA (Analysis of Variance) F-value (Table 8.1). The second is Mutual Information [77] that measures the dependency between two random variables (Table 8.2) for a discrete target variable. The method is based on entropy estimation from k-nearest neighbors distances.

As expected, the latent space based on style classification task have a better scores. The advantage of the unsupervised TTS and the speaker classification techniques is that a modeling of style can be obtained without labels concerning the style. Compared to the speaker classification technique, the unsupervised TTS technique performs better in terms of mutual information and F-value.

Table 8.1. Analysis of Variance (ANOVA) F-value between embedding dimensions and style categories in ascending order (higher is better).

Unsup-TTS	Style	Speaker
2430	75931	417
6770	77273	746
7037	89641	791
9156	94692	828
9510	107553	1284
11456	160017	1386
15468	197631	1533
16015	208180	3139

8.5.2 Relationship between the Embeddings and Audio Features

In this analysis, we study the relationship between the embeddings and eGeMAPS feature set [33], presented in Chapter 4. This feature set was designed based on their potential to represent affective physiological changes in speech.

Table 8.2. Mutual information between embedding dimensions and style categories in ascending order (higher is better).

Unsup-TTS	Style	Speaker
0.44	1.36	0.25
0.71	1.41	0.33
0.79	1.47	0.33
0.81	1.49	0.33
0.97	1.55	0.39
0.97	1.74	0.41
1.06	1.8	0.42
1.08	1.86	0.5

This analysis investigates which features describe best the remaining variability in the data.

The procedure is the following:

- We approximate a linear function between each latent space and the audio feature space with ordinary least squares linear regression. We thus obtain a hyper-plan that approximate audio features from latent embeddings.
- Then we estimate the correlation between predictions and ground truth through the Absolute Pearson Correlation Coefficient (APCC).

In other words, we compute the APCC between each audio feature and the best possible hyper-plan, in terms of least squares, of each latent space. To summarize these results, Table 8.3 shows features with APCC > 0.5 in every latent space.

8.5.3 Dimensionality reduction of latent spaces

In this section, we investigate the use of dimensionality reduction of latent spaces previously computed. Reducing the latent spaces to two dimensions

Table 8.3. APCC values between the best possible hyper-plan of each latent space and audio features of the eGeMAPS feature set.

APCC	Unsup-TTS	Style	Speaker
F0 mean	0.76	0.82	0.63
F0 percentile20.0	0.75	0.81	0.62
F0 percentile50.0	0.79	0.86	0.67
F0 percentile80.0	0.69	0.73	0.52
mfcc2 mean	0.73	0.77	0.65
mfcc4 mean	0.73	0.77	0.61
F1 freq mean	0.61	0.71	0.52
F2 freq mean	0.58	0.68	0.52
F3 freq mean	0.64	0.71	0.57
Alpha Ratio V mean	0.60	0.65	0.55
Hammarberg Index V mean	0.58	0.63	0.52
Slope V 0-500 mean	0.89	0.91	0.72
mfcc2 V mean	0.78	0.82	0.68
mfcc4 V mean	0.77	0.80	0.63

will enable the possibility to design an interface that allows its visualization and its relationship with audio features.

To that aim, we use three different algorithms of dimensionality reduction: Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP). We then perform the same procedure of regression as in Section 8.5.2 to obtain APCCs between each audio feature of Table 8.3 and the best possible hyper-plan, in terms of least squares, of each reduced latent space.

The steps are the following:

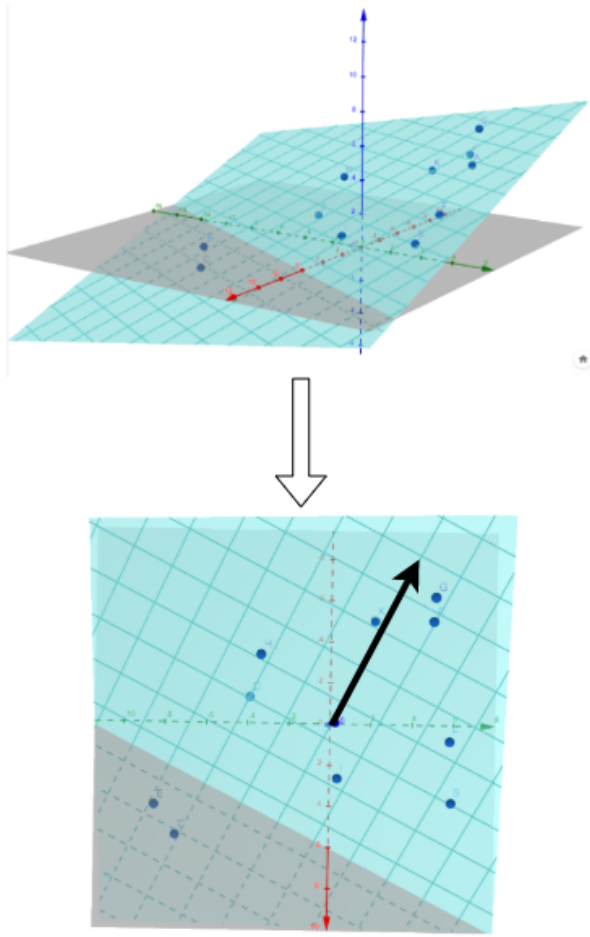

Figure 8.3. Gradient of the hyper-plan corresponding to the greatest slope.

- The mel-spectrogram is encoded to a vector of length 8 that contains expressiveness information. This vector is computed for each utterance of the dataset. We thus obtain an ensemble of 8D vectors.
- Dimensionality reduction is used to have an ensemble of 2D vectors instead. Figure 8.4 shows a scatter plot of these 2D points. These points

have a color corresponding to the style. That way we can assess that points located in some region of the space correspond to a specific style.

- Then a trend is extracted for each audio feature. As an example, $F0_{mean}$ is computed for each utterance of the dataset. We obtain therefore a $F0_{mean}$ corresponding to each 2D-points (x, y) of the scatter plot.
- We approximate the plan $F0 = f(x, y) = ax + by + c$.
- To assess that this plan $f(x, y)$ is a good approximation of $F0_{mean}$, implying a linear relation between a direction of the space and $F0_{mean}$, we compute the correlation between the approximations $f(x, y)$ with the ground truth values of $F0_{mean}$.
- If we compute the gradient of the plan (which is in fact (a, b)), we have the direction of the greatest slope, that is plotted in black in Figure 8.3.

Table 8.4. APCC average for each pair (task, dimensionality reduction algorithm).

	APCC
Unsup-TTS - PCA	0.564
Unsup-TTS - t-SNE	0.422
Unsup-TTS - UMAP	**0.614**
Style - PCA	0.480
Style - t-SNE	0.366
Style - UMAP	0.607
Speaker - PCA	0.512
Speaker - t-SNE	0.480
Speaker - UMAP	0.549

Table 8.4 shows the average APCC for each pair (task, dimension reduction algorithm). For each pair, the gradients of hyper-plans approximating audio features were computed. The higher average APCC corresponds to the pair (Unsup-TTS, UMAP). The direction of the gradients corresponds to the direction of the highest variation of a feature in the space. Figure 8.4 shows the reduced embeddings of all utterances of the dataset and the directions of the

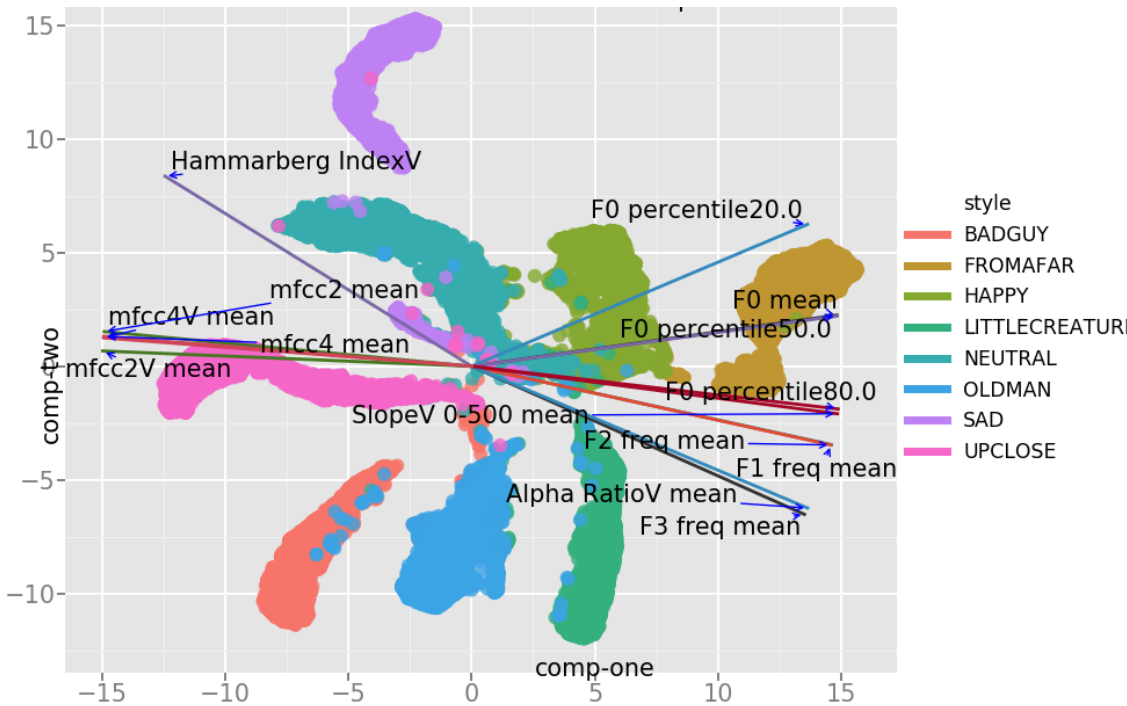

Figure 8.4. Reduced latent space with directions of feature gradients. The latent space is obtained by training a mel-spectrogram encoder together with a Sequence-to-Sequence TTS system. Then dimensionality reduction (in this case UMAP) is used for visualization. The clusters with different colors correspond to different speech styles. The straight lines indicate the directions of greatest variation of an acoustic feature obtained by linear regression.

gradients for the pair (Unsup-TTS, UMAP).

This representation is useful in a perspective of developing an interface for controllable speech synthesis system on which are represented the trends of audio features in the space.

8.6 Latent Space of Continuous Expressiveness Variability

In this section, we study the control of the Unsup-TTS system, trained on a dataset for a continuous control. The dataset is the Blizzard 2013 dataset mentioned in Chapter 3 based on audiobooks read by a female speaker containing a great variability in vocal expressions and therefore in acoustic features.

8.6.1 Quantitative Analysis

Distortion analysis: a comparison with typical seq2seq

To compare the synthesis performance of the proposed method with a typical seq2seq method, we compare objective measures used in expressive speech synthesis. These measures compute an error between acoustic features of a reference and a prediction of the model. There exist different types of objective measures that intend to quantify the distortion induced by a system on audio quality or prosody. In this work, we use the following objective measures:

- Mel Cepstral Distortion (MCD) [53] measuring speech quality: $\text{MCD}_K = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{k=1}^{K}\left(c_{t,k} - c'_{t,k}\right)^2}$
- Voiced Decision Error (VDE) [69]: $\text{VDE} = \frac{\sum_{t=0}^{T-1} 1[v_t \neq v'_t]}{T}$
- F_0 MSE measuring a distance between F_0 contours of prediction and ground truth: $\text{F0\_MSE} = \frac{1}{T}\sum_{t=0}^{T-1}\left(F_{0t} - F'_{0t}\right)^2$
- lF0 MSE, similar to previous one in logarithmic scale: $\text{lF0\_MSE} = \frac{1}{T}\sum_{t=0}^{T-1}\left(\log F_{0t} - \log F'_{0t}\right)^2$

Some works use DTW to align acoustic features before computing a distance. The problem with this method is that it modifies the rythm and speed of the sentence. However computing a distance on acoustic features that are shifted completely distorts the results, therefore, it is needed to apply a translation on acoustic features and take the smallest possible distance. We thus report

measures with DTW and with shift only in Table 8.5 for the original DCTTS and Table 8.6 for the proposed Unsupervised version of DCTTS.

Table 8.5. Objective measures for the typical TTS system.

	MCD	VDE	lF0_MSE	F0_MSE
DTW	9.974	0.015	0.436	1219.129
shift	13.332	0.236	6.284	9481.103

Table 8.6. Objective measures for the proposed Unsupervised TTS system.

	MCD	VDE	lF0_MSE	F0_MSE
DTW	9.624	0.010	0.312	957.290
shift	12.676	0.218	5.839	8931.600

Feature selection

To visualize acoustic trends, it would be useful to have a small number of features that gives a good overview. To extract a subset of the list, we apply a feature selection with a filtering method based on Pearson's correlation coefficient. The idea is to investigate correlations between audio features themselves to exclude redundant features and select a subset.

The steps are the following:

- features are sorted by APCC in decreasing order;
- for each feature, APCC with previous features are computed;
- if the maximum of these $inter - features - APCCs > 0.8$, the feature is eliminated;
- finally, only features that have a $prediction - APCC > 0.3$ are kept.

These limits are arbitrary and can be changed to filter more or less features from the list.

In Table 8.7, we show the results of the APCCs for Blizzard dataset and show the plot of gradients. It can be noted that F_0 median is the most predictable feature from the latent space. The feature selection method highlight a set of 17 diverse features that have an APCC > 0.3.

Table 8.7. APCC values between the best possible hyper-plan of the latent space and audio features of the eGeMAPS feature set.

	APCC
F0 percentile50.0	0.723
mfcc1V mean	0.620
mfcc1 mean	0.555
logRelF0-H1-A3 mean	0.493
mfcc4V mean	0.492
HNRdBACF mean	0.483
F1amplitudeLogRelF0 mean	0.473
slopeV0-500 mean	0.420
StddevVoicedSegmentLengthSec	0.389
F3amplitudeLogRelF0 stddevNorm	0.361
mfcc2V mean	0.360
hammarbergIndexV mean	0.356
mfcc1V stddevNorm	0.351
loudness meanFallingSlope	0.350
loudness percentile20.0	0.341
loudness meanRisingSlope	0.323
F1frequency mean	0.318

8.6.2 Qualitative Analysis

In this section, we follow the same procedure as in Section 8.5.3 to compute the directions in which acoustic features increase with a linear approximation. In Figure 8.5, we show a scatter plot of the reduced latent space with the feature gradients. Each point corresponds to one utterance encoding and reduced to two dimensions. The color of these points is mapped to the values of an acoustic feature to be able to visualize how the gradients are linked to the evolution of the acoustic features. Two examples are shown for F_0 median and standard deviation of voiced segment length, i.e., the duration of voiced sounds which is linked to the speaking rate.

We can observe that the direction of the gradients follows well the general trend of the corresponding acoustic feature. As the correlation values indicate, F_0 median has an evolution closer to a linear evolution in the direction of the gradient rather than for voiced segment lengths standard deviation.

8.7 Conclusions

This chapter presents a methodology to build latent spaces related to style/emotion in speech and visualize them along with their relationships with important audio feature for a purpose of controllable speech synthesis.

To that aim, we compared three latent spaces computed by training deep learning-based systems on three different tasks. We then examined the potential of these latent spaces for style classification to confirm that they contain useful information for representing style.

We then studied relationships between each latent space and audio features to obtain a sense of the impact of audio features on style expressed. This analysis consisted in an approximation of audio features from embeddings by linear regression. The accuracy of approximations was then evaluated in terms of correlations with ground truth.

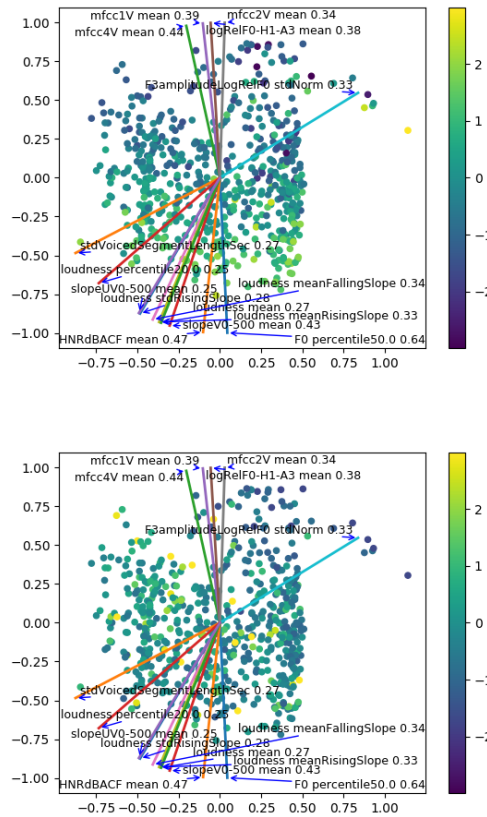

Figure 8.5. Reduced latent space with directions of gradients of features. The color of each point is the value of F0 median (top) and Voiced segment lengths standard deviation (bottom).

The gradient of these linear approximations are computed to extract the information of variations of audio features in speech. By visualizing these gradients along with the embeddings, we observe the trends of audio features in the latent space.

Chapter 9

A Proof of Concept: Integration in a Web Interface

Contents

This chapter is partially based on the following publication:

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "ICE-Talk: an Interface for a Controllable Expressive Talking Machine". In: *Proc. Interspeech 2020*. 2020, pp. 482–483

This chapter is part of the last of the 4 main tasks of the thesis work plan described in Section 1.2, i.e., controlling an expressive speech synthesis system with information extracted from a Deep Learning architecture.

To make things practical, the results of previous chapter are used to implement an interface allowing a user to input a text, and select a region of a latent space representing expressiveness.

ICE-Talk: "an Interface for a Controllable Expressive Talking Machine" is an open source[1] web-based Graphical User Interface (GUI) that allows the use of a TTS system with controllable parameters via a text field and a clickable 2D plot. It enables the study of latent spaces for controllable TTS. Moreover it is implemented as a module that can be used as part of a Human Agent interaction.

From this interface, an adaptation was developed to implement a perceptual experiment aiming to assess the controllability of such a system. This experiment studies if a user is able to find a sample with an expressiveness similar to a reference inside the 2D interface.

9.1 Introduction and Motivations

Speech Synthesis is an important component of Human-Robot Interaction. However as of today, expressiveness in speech generated by TTS systems is under-explored in such interactions. The reason is the difficulty of accessing the variables controlling speech expressiveness in a deep learning-based TTS system.

To tackle this problem we propose a tool allowing the control of these variables through a graphical interface, thus contributing to the democratization of the use of DL-based TTS systems in HAI applications. The goal of this tool is to be generic enough to be connected in any HAI application.

[1] https://github.com/noetits/ophelia

This interface allows for the control over the synthesis parameters of a DL-based model through its latent space directly and intuitively in a graphical way. It therefore allows the implementation of several interesting applications and experiments such as listening tests for the evaluation of such systems thanks to easy prototyping of experiments. An example of experiment is available. This tool also enables the possibility of studying the impact of expressive synthesized speech in Human-Robot interaction.

9.2 Related Work

As of today, there are some web interfaces allowing the use of DL TTS models[2]. They make it possible to write text, that is sent to the model and get the synthesized speech as an audio object that one can listen. The text is therefore the only control variable that we can access.

Recently, an interface[3] that allows the selection of a reference audio file and synthesize speech from text by imitating the voice of the reference [45] was developed. It is however not possible to interact with a latent space representing acoustic variability. In this chapter, we describe a web interface capable of visualizing and exploring a space of voice expressiveness and synthesize corresponding expressive speech.

9.3 Description of ICE-Talk

9.3.1 System architecture

Figure 9.1 depicts the different components of the system architecture. It is constituted of a DL unsupervised TTS model trained on an expressive dataset (see Section 9.3.2). To make the model available as a web service and communicate information of text, audio and style between the web interface and the TTS model, the Falcon Web framework[4] is used. Falcon bridges the gap

[2] https://github.com/keithito/tacotron
[3] https://github.com/CorentinJ/Real-Time-Voice-Cloning
[4] https://falcon.readthedocs.io/en/stable/

between a python code and a web interface, allowing the use of Deep Learning frameworks through a web application.

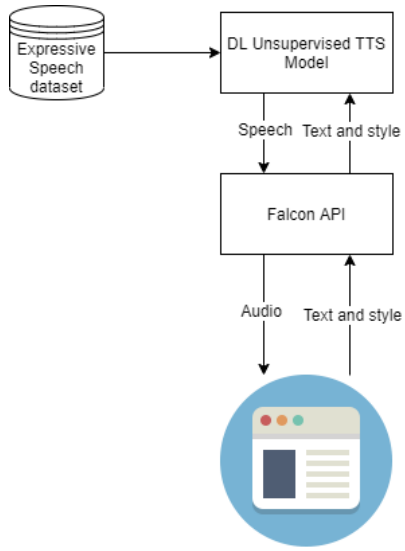

Figure 9.1. System architecture.

9.3.2 Deep Learning Unsupervised Model

As a use case, we use a modified version of DCTTS [93], a state-of-the art Deep-Learning Sequence-to-Sequence (seq2seq) model with an output controllable through a Latent Space designed to represent variations in voice style as described in Chapter 8.

A TTS seq2seq model typically consists of an encoder-decoder structure. Text is encoded as a latent representation that is then decoded with an attention based decoder to predict a mel-spectrogram later inverted to an audio waveform.

To obtain a voice style representation, a mel-spectrogram encoder is added. It consists of a stack of 1D convolutional layers, followed by an average pool-

ing. This operation ensures to obtain time-invariant information. It can thus contain information about statistics of prosody such as pitch average, average speaking rate, but not a pitch evolution.

9.3.3 Web Interface

The interface, shown in Figure 9.2, contains a 2D representation of a latent space which is an internal representation of the data distribution by the network. This 2D representation is obtained via a dimensionality reduction applied to the highly dimensional latent space of the system. The interface also contains a text box for the system's input and an audio player for the system's output.

The latent space represents the distribution of some control parameters (the expressiveness for instance) of the output speech, and is obtained after training. By writing a text and clicking on a point on the 2D space, an audio signal is generated with the parameters values corresponding to the point clicked on. The web interface is implemented in HTML5 and javascript to use the service.

There are several possibilities for dimensional reduction : UMAP, PCA or t-SNE. The click of the mouse is detected using javascript in pixels coordinates and mapped to the reduced data space.

Then Nearest Neighbour regression is used to compute the 2D data point, and a lookup table gives the corresponding 8D point of the latent space. The text and the 8D vector are fed to the model that generates the sentence and save it into an audio file. The audio file is then served and played as an HTML5 audio object.

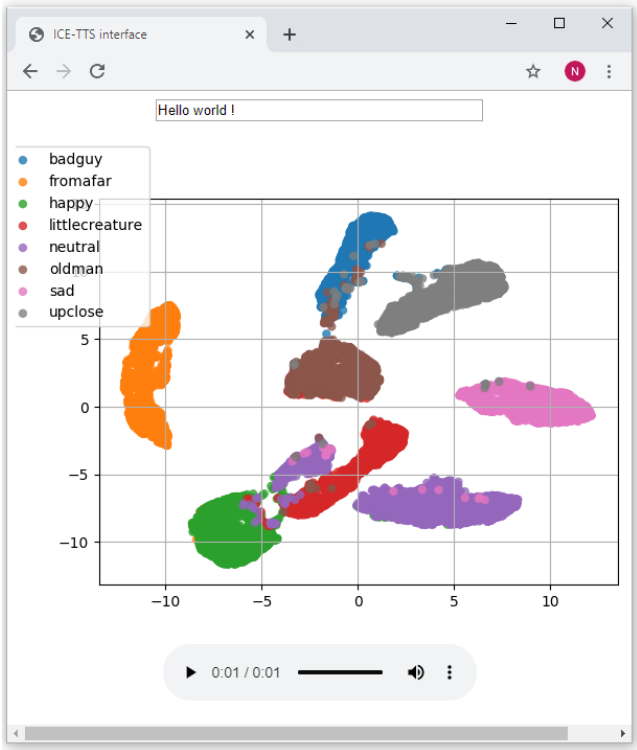

Figure 9.2. ICE-Talk web interface. It is constituted of a text field, an image representing a latent space of vocal variability contained in the dataset, and an audio player to listen to the synthesized utterance.

9.4 Perceptual Experiment

9.4.1 Methodology

An experiment was designed to assess the extent to which participants would be able to produce a desired expressiveness for a synthesized utterance, i.e., a methodology for evaluating the controllability of the expressiveness.

For this purpose, participants were asked to use the 2D interface to produce the same expressiveness as in a given reference. We assume that if a participant is

able to locate in the space the expressiveness corresponding to the reference, it means he is able to use this interface to find the expressiveness he has in mind.

The experiment contains two variants: in the first, the text of the reference and 2D space sentences are the same, while in the second, they are different. In the first one the participant can rely on the intonation and specific details of a sentence while in the second, he has to use a more abstract notion of expressiveness of a sentence.

The experiment is designed to avoid choosing a set of different characteristics or style categories, and letting the participant of the experiment judge how close the vocal characteristics of a synthesized sentence is to a reference.

The procedure for preparing the experiment is as follows:

- The model trained with Blizzard2013 dataset mentioned in Chapter 3, is used to generate a latent space with continuous variations of expressiveness as presented in Section 8.6.
- In the 2D interface, we sample a set of points inside the region of the space in which the dataset points are located. The limits of the rectangle are defined by projecting sentences of the whole dataset in the 2D space with PCA and selecting x_{min}, x_{max}, y_{min}, y_{max} of all points. In other words, we use the smallest rectangle containing the dataset points. We use a resolution of 100 for x and y axes, making a total of 10000 points in the space.
- This set of 2D points is projected to the 8D latent space of the trained unsupervised model with inverse PCA. The 8D vectors will then be fed to the model for synthesis.
- 5 different texts are used to synthesize the experiment materials. This makes a total of 50000 expressive sentences synthesized with the model.

The listening test was implemented with the help of turkle[5] mentioned in Chapter 6 and 7. We can ask questions with an HTML template that includes in this case an interface implemented in HTML/javascript.

During the perceptual experiment, a reference sentence coming from the 50000 sentences is provided to the participants. We provide the interface allowing a participant to click in the latent space and choose what is the point that is in his opinion the closest to the reference in terms of expressiveness.

The instructions shown to participants are the following:

- First, before the experiment, to illustrate what kind of task it will contain and familiarize you with it, here is a link to a demo interface: https://jsfiddle.net/g9aos1dz/show
- You can choose the sentence and you have a 2D space on which you can click. It will play the sentence with a specific expressiveness depending on its location.
- Familiarize yourself with it and listen to different sentences with a different expressiveness.
- Then for the experiment, use headphones to hear well, and be in a quiet environment where you will not be bothered.
- You will be asked to listen to a reference audio sample and find the red point in the 2D space that you feel to be the closest in expressiveness.
- Be aware that expressiveness varies continuously in the entire 2D space.
- You can click as much as you like on the 2D space and replay a sample. When you are satisfied with your choice, click on submit.
- There are two different versions, in the first one, the sentence is the same in the reference and in the 2D space. In the second, they are not. You just need to select the red point that in your opinion has the closest expressiveness.
- It would be great if you could do this for a set of 15 samples in each level. You can see your evolution on the page.

[5] https://github.com/hltcoe/turkle

A number of 25 and 26 people participated in variants 1 and 2 of the experiment, respectively. We collected a total of 488 and 326 answers.

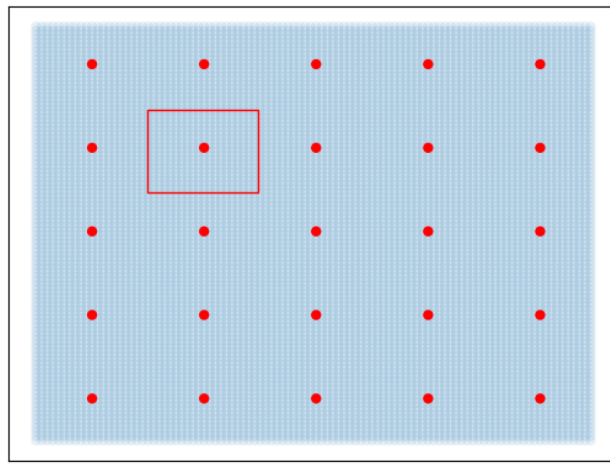

Figure 9.3. 2D space fractionned in a 5x5 grid for the perceptual experiment. The red points are the possible positions of the reference in the space, the red rectangle is the selected case.

9.4.2 Evaluation

Controllability score To quantify how well the participants are able to produce a desired expressiveness, we compute an average euclidean distance between the selected point and its true location.

Inspired by the omnipresent 5-point scales in the field of perceptual assessment, such as MOS tests, we choose to discretize the 2D space in a five-by-five grid, as shown in Figure 9.3. Indeed a continuous scale could be overwhelming for participants and let them unsure about their decision. The unit of distance

is that between a red point and its neighbour along the horizontal axis.

We use a random baseline to assess the level of a non-controllability of the system in terms of expressiveness. In other words, if a participant is not able to distinguish the differences in expressiveness of different samples, we assume that he would not be able to select the correct location of the expressiveness of the reference, and would answer randomly.

Results and Discussion Figure 9.4 shows the distributions of the distances between participant answers and true location of references in the 2D space. The two variants (with same text and different text) are on the left and the random baseline is on the right. The average distances with 95% confidence intervals of the three distributions are respectively: 0.908 ± 0.083, 1.448 ± 0.103 and 2.314 ± 0.007.

The second version was considered much more difficult by participants. For the first task, it is possible to listen to every detail of the intonation to detect if the sentence is the same. That strategy is not possible for the second one in which only an abstract notion of expressiveness has to be imagined.

Also, the speech rate is more difficult to compare between two different sentences than for the same sentence. Especially when the number of syllables is a lot different, it is more difficult to compare the melody and the rhythm of the sentences. The cues mentioned by participants include intonation, tonic accent, speech rate and rhythm.

We can see in Figure 9.5 that, over time, participants are progressively more constant in the duration and with a lower median duration. Outliers were discarded because they were too far from the distribution. The maximum is above 17500 seconds. We believe these outliers are due to a break taken by participants during the test. Also, the means are influenced by these outliers, and are therefore not plotted in the figure.

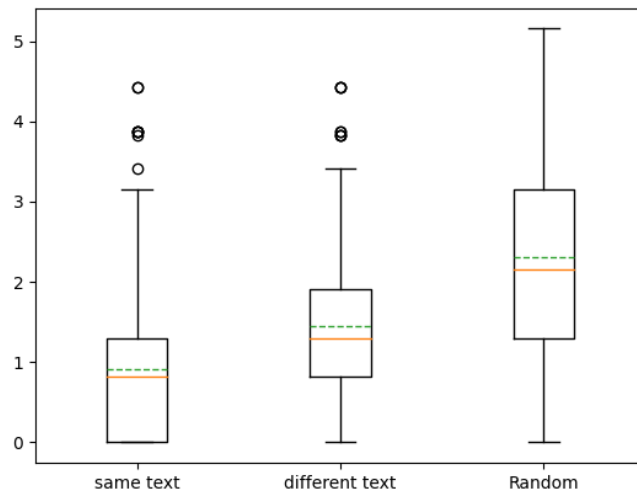

Figure 9.4. Boxplots of the distances between participant choices and true location of the reference (lower is better). The green line corresponds to the mean, and the orange line corresponds to the median. From left to right: 1) results of the first variant of the experiment for which the text synthesized is the same for the reference and the latent space, 2) results of the second variant for which the text synthesized is different for the reference and the latent space, 3) a random baseline

A least square linear regression on the medians shows that it decreases with a slope of -0.767 s/task for the first variant and -2.086 s/task for the second. The two-sided p-value for a hypothesis test whose null hypothesis is that the slope is zero are respectively 0.21 and 0.0004. We can therefore reject the null hypothesis in the second case but not in the first.

Participants mentioned that they were more confident in both tasks after several samples. They could guess where they have to search. They could establish a strategy as they understood how the space was structured. Therefore they felt like it was easier and could make a choice faster because they

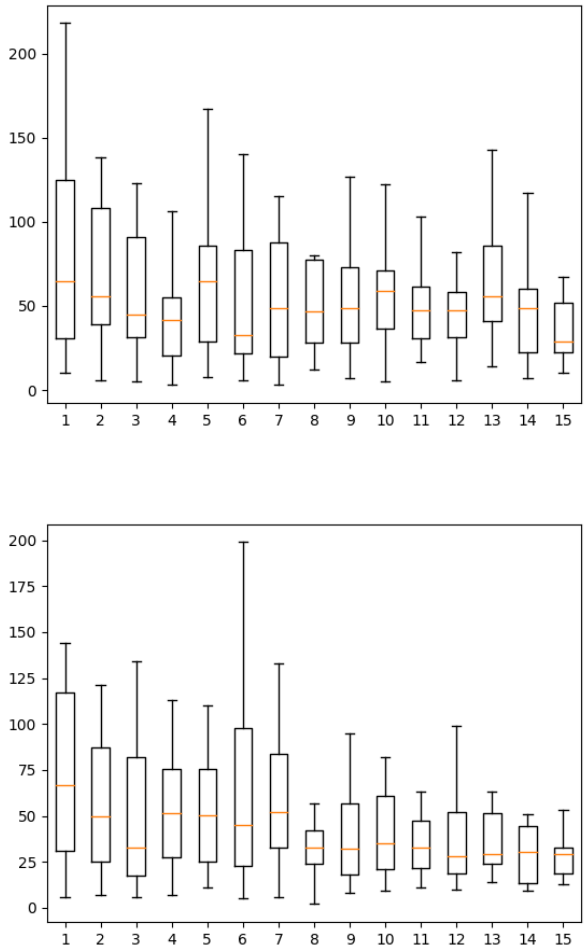

Figure 9.5. Boxplots of the durations for participant to answer by index until the 15th answer of participants for variant 1 (top) and 2 (bottom) of the experiment.

hesitated less.

However, the evolution of average scores, depicted in Figure 9.6, does not seem to improve or decline over time. A least square linear regression on the average scores show slopes close to zero for both variant 1 and 2 (respectively -0.005 and 0.0001 s/task). The two-sided p-value for a hypothesis test whose null hypothesis is a zero-slope line are respectively 0.496 and 0.930. It indicates strong evidence that the slope is zero, i.e., the evolution of average scores remains stable.

9.5 Conclusions and Future Works

We presented an innovative interface for Controllable Expressive TTS called ICE-Talk that shows a proof of concept of research results previously presented at Interspeech. It is open source and ready to be used with available pre-trained models.

As a perspective, other models such as a multi-speaker TTS[6] could be integrated to be able to generate speech from different speakers, not based on references but on the latent space describing speaker characteristics.

A perceptual experiment was designed to evaluate the controllability of a Controllable Expressive TTS model. For that purpose, a set of samples were synthesized by discretizing the 2D reduced latent space with a set of five different sentences. This 2D space was then used to ask participants to search the location corresponding to the expressiveness of an audio reference. To reduce the number of possible answers, the 2D space was segmented in a five by five grid. An average distance can then be computed and compared to a random baseline. Two variants of the task were presented to participants: in the first one, the transcription of the audio reference and samples contained in the 2D space are the same, while in the second, they are different. Results show that the average distance is lower for the first task than for the second, and that they are both lower than the random baseline.

[6] https://github.com/CorentinJ/Real-Time-Voice-Cloning

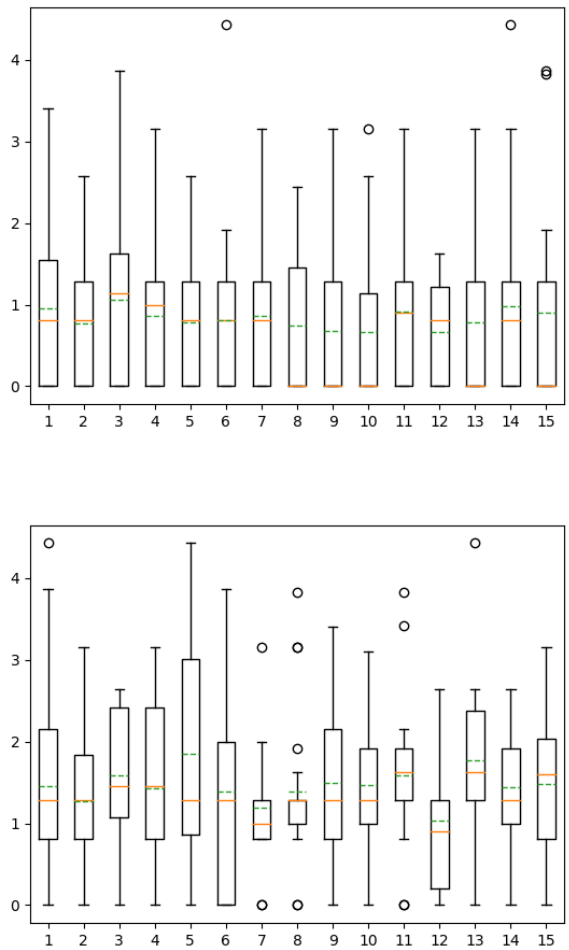

Figure 9.6. Boxplots of the distances between participant choices and true location of the reference by index until the 15th answer of participants for variant 1 (top) and 2 (bottom) of the experiment.

Chapter 10

Conclusions

This section summarizes the contributions presented in the thesis and proposes a set of possible directions of the field and industrial applications.

An emotional speech dataset As of today, there are different kinds of speech datasets available online. For example, a good source of speech data are audiobooks for which the transcriptions are systematically available.

Other datasets come from research projects. Some are designed for speech generation purpose while others are designed for speech analysis (ASR, emotion in speech, etc.). Compared to speech generation datasets, the latter often consist of conversational setups and contain overlaps in speech, i.e., actors speaking at the same time, as well as noise. This is the reason why they cannot be used for emotional speech synthesis. In most cases, these datasets contain enough data for the use of deep learning algorithms but, either they are not suited for synthesis purpose, or they are poor in expressiveness. Or, on the contrary, they offer emotionally rich content in high quality, but in a limited amount.

For this project, we collected a speech dataset called EmoV-DB Database rich in expressiveness, suited for speech generation purpose and contains enough data to use deep learning algorithms. The collected data are recordings of male and female actors speaking in English. The actors were recorded in two different anechoic chambers of the Northeastern University campus. Each actor was asked to utter a subset of the sentences from the CMU-Arctic database for English speakers. The dataset covers 5 classes (amusement, anger, disgust, sleepiness and neutral) with the aim of building synthesis and voice transformation systems.

Emotion representation We built a system able to extract a representation of emotional expressiveness in speech. We examined whether a DNN trained in an ASR task learned useful features in predicting the dimensions of emotions (valence, arousal).

An ASR system is a mapping between audio features and text features. For an ASR to be accurate, it has to take into account the variability in speech and doing so, recover text information from all kinds of speaker identity, style etc. Therefore, an intermediate representation extracted by the system may contain useful information to predict emotion in speech.

The ASR model consists of a stack of 15 dilated convolutional layers and each layer consists of 128 gated convolutional units (GCU) with skip connection. This architecture is inspired by Wavenet [71]. The system was trained with MFCCs extracted from VCTK dataset [109].

The analysis on IEMOCAP dataset reveals that predicting valence and arousal with a linear model trained based on the GCUs' output outperforms a similar model based on the eGeMAPS [33] feature set.

Synthesis with Emotion adaptation In this study, we showed how to obtain an emotional TTS system by fine-tuning a neutral TTS system with a small emotional speech dataset. We studied the impact of this fine-tuning on the intelligibility of generated speech and the subjective perception of expressed emotion. A first model was trained with LJ-speech dataset, a dataset that consists of sentences uttered by a female speaker, with a total of 24 hours of speech. To obtain emotional TTS models by fine-tuning, we used a subset of EmoV-DB corresponding to one speaker and proceeded in two steps:

- fine-tuning the model with the neutral part of the subset;
- fine-tuning copies of the resulting model separately with subsets of each emotional categories.

To evaluate the first step, we used an objective measure of intelligibility proposed in [73] on 100 sentences synthesized with three systems: the first model trained on LJ-speech dataset, the model fine-tuned with the neutral part of the subset and a model trained only with the neutral part of the subset. The average accuracies are respectively 0.630 ± 0.042, 0.517 ± 0.048 and 0.004 ± 0.004.

These results suggest that the fine-tuning on a small subset tends to degrade the intelligibility only slightly. However, using just the small dataset leads to unintelligible speech-like sounds. This demonstrates that the fine-tuning is useful to generate more intelligible speech.

To evaluate the second step, we performed a subjective MOS test to measure the perception of the emotion for each emotional model. The results show that the emotions are perceptible although less intelligible.

An application to Acoustic Laughter Synthesis Researchers of the ISIA laboratory have often successfully applied technologies of speech synthesis to the generation of laughter. This nonverbal expression is an important component for conveying emotion expressiveness in synthetic speech. This contribution is an application of Seq2seq technique to audio laughter synthesis, leveraging fine-tuning from the speech synthesis task. In order to perform the adaptation, we used AmuS database which contains amused speech and laughters from the same speaker.

We studied two variations of the system and compared it to previous methods based on HMM synthesis to assess a higher synthesis quality. We showed that the integration of MelGAN as a waveform corrector improves the performance of the synthesis.

Impact of controlling the intensity of emotional categories We developed a multi-style TTS system which has the possibility to control the intensity of style categories. We implemented a modified version of DCTTS that uses a style category encoding as input. Then, a one-hot encoding is used for the training. Finally, at the synthesis stage, we can modify the intensity of a style category by interpolating between the one-hot encodings of neutral and one of the styles.

To analyze the impact of control parameters on the perception of the generated sentence, we carried out a listening test in which we asked participants to compare pairs of generated samples and select the one conveying a given speech style with more intensity. The results of these tests are represented in a heatmap to visualize the relationship between the perception of intensity and the control variable. It showed that despite being trained with only discrete

categories of data, the network is capable of generating intermediate intensity levels between neutral and a given speech style.

Expressive synthesis and Interpretation of latent spaces The aim is to investigate ways to represent expressiveness in speech without the need of any labels that are difficult and expensive to collect. Moreover, it is interesting to have an interpretable representation in order to build controllable speech synthesis systems with understandable and predictable behaviour.

We presented a methodology to obtain latent spaces, visualize them and analyze their relationship with audio features by correlation. We then described a way to visualize the influence of embeddings variation on audio features.

We compared three encoding techniques of mel-spectrogram. The principle is to encode the mel-spectrogram by training a deep learning algorithm on a given task. The encoding is used for the following tasks:

- style category classification;
- speaker classification;
- unsupervised expressive Text-to-Speech synthesis.

For each encoding technique, the mel-spectrogram is encoded to a vector of length 8 that contains expressiveness information. This vector is computed for each utterance of the dataset to obtain a collection of embeddings. Dimensionality reduction is then performed to have a 2D-space that can be visualized. We observed that points in a given region of the resulting space correspond to a given style.

To interpret that space, we analyzed the trend of variation of some audio features. For this purpose, each audio feature, i.e. the F0 mean, is computed for each utterance of the dataset. Each F0 mean corresponds to one of the points in the 2D-space. A plane $F0_{mean} = f(x, y) = ax + by + c$ is then approximated. We evaluate if this plane $f(x, y)$ is a good approximation of $F0_{mean}$. If it is, it means that going in a certain direction of the space will increase $F0_{mean}$ linearly. We verified this by computing the correlation between the approximations $f(x, y)$ with the real values of $F0_{mean}$. The gradient of the plane is ($\frac{\partial f}{\partial x} = a$, $\frac{\partial f}{\partial y} = b$) and corresponds to the direction of its greatest slope.

It thus gives the direction of the variation of the audio feature approximated by the plane.

The embedding space trained to condition a TTS system has several advantages: it does not require any label about the style and it is already in use in a TTS system.

A further analysis of the latter system was carried out with a speech dataset that contains a continuous expressiveness variability rather than a set of style categories. We identified acoustic characteristics with important variations in the dataset using a feature selection method, and we assessed the quality of the synthesized speech.

A proof of concept and assessment of controllability We showed how the research results of this thesis can be implemented in a demonstration of expressive speech synthesis. This demonstration can be executed via a web interface on a standard computer. This demonstration enables easier access to the technologies in this field.

From this interface, an adaptation was developed to implement a perceptual experiment aiming to evaluate the controllability of a Controllable Expressive TTS model. This experiment studies if a user is able to find a sample with a similar expressiveness inside the 2D interface. Two variants of this task were presented to participants: in the first one, the transcription of the audio reference and samples contained in the 2D space are the same, while in the second, they are different. Results show that the average distance between participant choices and ground truth is lower for the first task than for the second, and that they are both lower than a random choice.

Perspectives

In Chapter 9, we present a 2D interface in which we can explore a space of expressiveness. It could be interesting to investigate ways to control more vocal characteristics, and independently when it is consistent and possible. Several types of controls could be investigated depending on the nature of the variables. For some variables, the control could consist of a set of choices, e.g.,

male/female, or a list of speaker identities.

We also could imagine to have two separate 2D spaces. One would be dedicated to a speaker identity, i.e., a space organizing voice timbers. And the second would, e.g., the 2D space of expressiveness presented in Chapter 8 and integrated in an interface in Chapter 9. This kind of application needs frameworks able to *disentangle* speech characteristics and factorize information corresponding to different phenomena, such as phonetics, speaker characteristics and expressiveness in the generated speech.

In the idea of having more and more general systems, the research results of this thesis that focus on English language could be adapted to obtain a system able to work with several languages. This could be considered as one more aspect of speech that needs to be *factorized* with others mentioned in previous paragraph.

There are also possibilities of controlling the evolution of speech characteristics inside a sentence, referred to as *fine-grained* control that could be interesting to investigate. Currently, this aspect is mostly present in *prosody transfer* task and is not subject to a control involving a human choosing what intonation, tonic accent or voice quality he would like to hear at different parts of a sentence. The difficulty would be to select the relevant characteristics that a sound designer would want to control and design an intuitive interface to control them.

The different possibilities in this area would be interesting for, e.g., video games producer for the development of virtual characters with expressive voices, for animation movies, synthetic audiobooks, or in the advertisement sector.

Appendix A

Publications related to this thesis

A.1 Regular Papers Referenced by Scopus

- Noé Tits, Fengna Wang, Kevin El Haddad, Vincent Pagel, and Thierry Dutoit. "Visualization and Interpretation of Latent Spaces for Controlling Expressive Speech Synthesis through Audio Analysis". In: *Proc. Interspeech 2019*. 2019, pp. 4475–4479. DOI: 10.21437/Interspeech.2019-1426. URL: http://dx.doi.org/10.21437/Interspeech.2019-1426
- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Laughter Synthesis: Combining Seq2seq Modeling with Transfer Learning". In: *Proc. Interspeech 2020*. 2020, pp. 3401–3405. DOI: 10.21437/Interspeech.2020-1423. URL: http://dx.doi.org/10.21437/Interspeech.2020-1423
- Noé Tits. "A Methodology for Controlling the Emotional Expressiveness in Synthetic Speech - a Deep Learning approach". In: *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. 2019, pp. 1–5. DOI: 10.1109/ACIIW.2019.8925241
- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Exploring Transfer Learning for Low Resource Emotional TTS". in: *Intelligent Systems and Applications*. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 52–60. ISBN: 978-3-030-29516-5
- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Emotional Speech Datasets for English Speech Synthesis Purpose: A Review". In: *In-*

telligent Systems and Applications. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 61–66. ISBN: 978-3-030-29516-5

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Neural Speech Synthesis with Style Intensity Interpolation: A Perceptual Analysis". In: *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '20. Cambridge, United Kingdom: Association for Computing Machinery, 2020, pp. 485–487. ISBN: 9781450370578. DOI: 10.1145/3371382.3378297. URL: https://doi.org/10.1145/3371382.3378297

A.2 Papers in International Conferences with Peer Review

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "ASR-based Features for Emotion Recognition: A Transfer Learning Approach". In: *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 48–52. URL: http://aclweb.org/anthology/W18-3307
- Jean-Benoit Delbrouck, Noé Tits, Brousmiche Mathilde, and Stéphane Dupont. "A Transformer-based joint-encoding for Emotion Recognition and Sentiment Analysis". In: *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*. Seattle, USA: Association for Computational Linguistics, July 2020, pp. 1–7. URL: https://www.aclweb.org/anthology/2020.challengehml-1.1
- Jean-Benoit Delbrouck, Noé Tits, and Stéphane Dupont. "Modulated Fusion using Transformer for Linguistic-Acoustic Emotion Recognition". In: *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1–10. URL: https://www.aclweb.org/anthology/2020.nlpbt-1.1

- Kevin El Haddad, Noé Tits, and Thierry Dutoit. "Annotating Nonverbal Conversation Expressions in Interaction Datasets". In: *Proceedings of Laughter Workshop 2018*. Sept. 2018

A.3 Book Chapters

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "The Theory behind Controllable Expressive Speech Synthesis: A Cross-Disciplinary Approach". In: *Human-Computer Interaction*. IntechOpen, 2019. DOI: 10.5772/intechopen.89849. URL: http://dx.doi.org/10.5772/intechopen.89849
- Kevin El Haddad, Noé Tits, Ella Velner, and Hugo Bohy. "Cozmo4Resto: A Practical AI Application for Human-Robot Interaction". In: *15th International Summer Workshop on Multimodal Interfaces*. 2019, p. 12. ISBN: 978-605-9788-33-5

A.4 Abstracts/Demos in International Conferences with Peer Review

- Noé Tits, Kevin El Haddad, and Thierry Dutoit. "ICE-Talk: an Interface for a Controllable Expressive Talking Machine". In: *Proc. Interspeech 2020*. 2020, pp. 482–483
- Adaeze Adigwe, Tits Noé, El Haddad Kevin, Ostadabbas Sarah, Dutoit Thierry, "The Emotional Voices Database: Towards Controlling the Emotion Dimension in Voice Generation Systems". In: *"International Conference on Statistical Language and Speech Processing"* , Mons, Belgique (2018)

Bibliography

[1] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. "Expressive Speech Synthesis via Modeling Expressions with Variational Autoencoder". In: *Proc. Interspeech 2018*. 2018, pp. 3067–3071. DOI: 10.21437/Interspeech.2018-1113. URL: http://dx.doi.org/10.21437/Interspeech.2018-1113.

[2] Diane Mayasari Alamsaputra, Kathryn J Kohnert, Benjamin Munson, and Joe Reichle. "Synthesized speech intelligibility among native speakers and non-native speakers of English". In: *Augmentative and Alternative Communication* 22.4 (2006), pp. 258–268.

[3] Sercan Ö Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. "Deep voice: Real-time neural text-to-speech". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 195–204.

[4] Tanja Bänziger, Marcello Mortillaro, and Klaus R Scherer. "Introducing the Geneva Multimodal expression corpus for experimental research on emotion perception." In: *Emotion* 12.5 (2012), p. 1161.

[5] Lisa Feldman Barrett. "The theory of constructed emotion: an active inference account of interoception and categorization". In: *Social cognitive and affective neuroscience* 12.1 (2017), pp. 1–23.

[6] Nicholas Becherer, John Pecarina, Scott Nykl, and Kenneth Hopkinson. "Improving optimization of convolutional neural networks through

parameter fine-tuning". In: *Neural Computing and Applications* (2017), pp. 1–11.

[7] Alan Black, Paul Taylor, Richard Caley, and Rob Clark. *The festival speech synthesis system.* 1998.

[8] Bajibabu Bollepalli, Jérôme Urbain, Tuomo Raitio, Joakim Gustafson, and Hüseyin Cakmak. "A comparative evaluation of vocoding techniques for hmm-based laughter synthesis". In: *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP).* IEEE. 2014, pp. 255–259.

[9] Sandrine Brognaux, Sophie Roekhaut, Thomas Drugman, and Richard Beaufort. "Train&Align: A new online tool for automatic phonetic alignment". In: *Spoken Language Technology Workshop (SLT), 2012 IEEE.* IEEE. 2012, pp. 416–421.

[10] Felix Burkhardt and Nick Campbell. "Emotional speech synthesis". In: *The Oxford Handbook of Affective Computing.* Oxford University Press New York, 2014, p. 286.

[11] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter F Sendlmeier, and Benjamin Weiss. "A database of German emotional speech". In: *Ninth European Conference on Speech Communication and Technology.* 2005.

[12] Felix Burkhardt and Walter F Sendlmeier. "Verification of acoustical correlates of emotional speech using formant-synthesis". In: *ISCA Tutorial and Research Workshop (ITRW) on speech and emotion.* 2000.

[13] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. "IEMOCAP: Interactive emotional dyadic motion capture database". In: *Language resources and evaluation* 42.4 (2008), p. 335.

[14] Carlos Busso, Srinivas Parthasarathy, Alec Burmania, Mohammed AbdelWahab, Najmeh Sadoughi, and Emily Mower Provost. "MSP-IMPROV: An acted corpus of dyadic interactions to study emotion perception". In: *IEEE Transactions on Affective Computing* 8.1 (2017), pp. 67–80.

[15] Houwei Cao, David G Cooper, Michael K Keutmann, Ruben C Gur, Ani Nenkova, and Ragini Verma. "CREMA-D: Crowd-sourced emotional multimodal actors dataset". In: *IEEE transactions on affective computing* 5.4 (2014), pp. 377–390.

[16] Rich Caruana. "Multitask learning". In: *Machine learning* 28.1 (1997), pp. 41–75. DOI: 10.1023/A:1007379606734.

[17] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 4960–4964.

[18] Jean-Benoit Delbrouck, Noé Tits, and Stéphane Dupont. "Modulated Fusion using Transformer for Linguistic-Acoustic Emotion Recognition". In: *Proceedings of the First International Workshop on Natural Language Processing Beyond Text*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1–10. URL: https://www.aclweb.org/anthology/2020.nlpbt-1.1.

[19] Jean-Benoit Delbrouck, Noé Tits, Brousmiche Mathilde, and Stéphane Dupont. "A Transformer-based joint-encoding for Emotion Recognition and Sentiment Analysis". In: *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*. Seattle, USA: Association for Computational Linguistics, July 2020, pp. 1–7. URL: https://www.aclweb.org/anthology/2020.challengehml-1.1.

[20] Thomas Drugman, Geoffrey Wilfart, and Thierry Dutoit. "A deterministic plus stochastic model of the residual signal for improved parametric speech synthesis". In: *arXiv preprint arXiv:2001.00842* (2019).

[21] Thierry Dutoit. *An introduction to text-to-speech synthesis*. Vol. 3. Springer Science & Business Media, 1997.

[22] Paul Ekman. "An argument for basic emotions". In: *Cognition & emotion* 6.3-4 (1992), pp. 169–200.

[23] Paul Ekman. "Basic Emotions". In: *Handbook of Cognition and Emotion*. Ed. by Tim Dalgleish and M. J. Powers. Wiley, 1999, pp. 4–5.

[24] Kevin El Haddad, Hüseyin Cakmak, Stéphane Dupont, and Thierry Dutoit. "An HMM Approach for Synthesizing Amused Speech with a Controllable Intensity of Smile". In: *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE. Abu Dhabi, UAE, 2015, pp. 7–11.

[25] Kevin El Haddad, Hüseyin Cakmak, Stéphane Dupont, and Thierry Dutoit. "Breath and repeat: An attempt at enhancing speech-laugh synthesis quality". In: *European Signal Processing Conference (EUSIPCO 2015)*. Nice, France, 2015.

[26] Kevin El Haddad, Stéphane Dupont, Nicolas d'Alessandro, and Thierry Dutoit. "An HMM-based Speech-smile Synthesis System: An approach for amusement synthesis". In: *International Workshop on Emotion Representation, Analysis and Synthesis in Continuous Time and Space (EmoSPACE)*. Ljubljana, Slovenia, 2015.

[27] Kevin El Haddad, Stéphane Dupont, Jérôme Urbain, and Thierry Dutoit. "Speech-laughs: An HMM-based Approach for Amused Speech Synthesis". In: *Internation Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*. Brisbane, Australia, 2015, pp. 4939–4943.

[28] Kevin El Haddad, Noé Tits, and Thierry Dutoit. "Annotating Nonverbal Conversation Expressions in Interaction Datasets". In: *Proceedings of Laughter Workshop 2018*. Sept. 2018.

[29] Kevin El Haddad, Noé Tits, Ella Velner, and Hugo Bohy. "Cozmo4Resto: A Practical AI Application for Human-Robot Interaction". In: *15th International Summer Workshop on Multimodal Interfaces*. 2019, p. 12. ISBN: 978-605-9788-33-5.

[30] Kevin El Haddad, Ilaria Torre, Emer Gilmartin, Hüseyin Çakmak, Stéphane Dupont, Thierry Dutoit, and Nick Campbell. "Introducing AmuS: The Amused Speech Database". In: *Statistical Language and Speech Processing*. Ed. by Nathalie Camelin, Yannick Estève, and Carlos Martín-Vide. Cham: Springer International Publishing, 2017, pp. 229–240. ISBN: 978-3-319-68456-7.

[31] H Elovitz, Rodney Johnson, Astrid McHugh, and J Shore. "Letter-to-sound rules for automatic translation of english text to phonetics". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.6 (1976), pp. 446–459.

[32] Daniel Erro, Inaki Sainz, Eva Navas, and Inma Hernaez. "Harmonics plus noise model based vocoder for statistical parametric speech synthesis". In: *IEEE Journal of Selected Topics in Signal Processing* 8.2 (2013), pp. 184–194.

[33] Florian Eyben, Klaus R Scherer, Björn W Schuller, Johan Sundberg, Elisabeth André, Carlos Busso, Laurence Y Devillers, Julien Epps, Petri Laukka, Shrikanth S Narayanan, et al. "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing". In: *IEEE Transactions on Affective Computing* 7.2 (2016), pp. 190–202.

[34] Florian Eyben, Martin Wöllmer, and Björn Schuller. "Opensmile: the munich versatile and fast open-source audio feature extractor". In:

Proceedings of the 18th ACM international conference on Multimedia. ACM. 2010, pp. 1459–1462.

[35] Saeed Gazor and Wei Zhang. "Speech probability distribution". In: *IEEE Signal Processing Letters* 10.7 (2003), pp. 204–207.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[37] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks". In: *Proceedings of the 23rd international conference on Machine learning.* 2006, pp. 369–376.

[38] Daniel Griffin and Jae Lim. "Signal estimation from modified short-time Fourier transform". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (1984), pp. 236–243.

[39] Mark G Hall, Alan V Oppenheim, and Alan S Willsky. "Time-varying parametric modeling of speech". In: *Signal Processing* 5.3 (1983), pp. 267–285.

[40] Gustav Eje Henter, Jaime Lorenzo-Trueba, Xin Wang, and Junichi Yamagishi. "Deep Encoder-Decoder Models for Unsupervised Learning of Controllable Speech Synthesis". In: *arXiv preprint arXiv:1807.11470* (2018).

[41] Sepp Hochreiter. "The vanishing gradient problem during learning recurrent neural nets and problem solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[42] Pierre-Edouard Honnet, Alexandros Lazaridis, Philip N. Garner, and Junichi Yamagishi. "The SIWIS French Speech Synthesis Database ? Design and recording of a high quality French database for speech synthesis". In: *Online Database* (2017).

[43] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. "Hierarchical Generative Modeling for Controllable Speech Synthesis". In: *arXiv preprint arXiv:1810.07217* (2018).

[44] Keith Ito. *The LJ Speech Dataset*. https://keithito.com/LJ-Speech-Dataset/. 2017.

[45] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. "Transfer learning from speaker verification to multispeaker text-to-speech synthesis". In: *Advances in neural information processing systems*. 2018, pp. 4480–4490.

[46] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. "Efficient Neural Audio Synthesis". In: (2018), pp. 2410–2419.

[47] Sri Karlapati, Alexis Moinet, Arnaud Joly, Viacheslav Klimkov, Daniel Sáez-Trigueros, and Thomas Drugman. "CopyCat: Many-to-Many Fine-Grained Prosody Transfer for Neural Text-to-Speech". In: *Proc. Interspeech 2020*. 2020, pp. 4387–4391. DOI: 10.21437/Interspeech.2020-1251. URL: http://dx.doi.org/10.21437/Interspeech.2020-1251.

[48] Jaebok Kim, Gwenn Englebienne, Khiet P. Truong, and Vanessa Evers. "Deep Temporal Models Using Identity Skip-Connections for Speech Emotion Recognition". In: *Proceedings of the 2017 ACM on Multimedia Conference*. MM '17. Mountain View, California, USA: ACM, 2017, pp. 1006–1013. ISBN: 978-1-4503-4906-2. DOI: 10.1145/3123266.3123353. URL: http://doi.acm.org/10.1145/3123266.3123353.

[49] Jaebok Kim, Gwenn Englebienne, Khiet P. Truong, and Vanessa Evers. "Towards Speech Emotion Recognition "in the Wild" Using Aggregated Corpora and Deep Multi-Task Learning". In: *INTERSPEECH*. 2017.

[50] Viacheslav Klimkov, Srikanth Ronanki, Jonas Rohnke, and Thomas Drugman. "Fine-Grained Robust Prosody Transfer for Single-Speaker Neural Text-To-Speech". In: *Proc. Interspeech 2019*. 2019, pp. 4440–4444. DOI: 10.21437/Interspeech.2019-2571. URL: http://dx.doi.org/10.21437/Interspeech.2019-2571.

[51] John Kominek and Alan W Black. "The CMU Arctic speech databases". In: *Fifth ISCA Workshop on Speech Synthesis*. 2004.

[52] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. "Multiplicative LSTM for sequence modelling". In: *arXiv preprint arXiv:1609.07959* (2016).

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

[54] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. "Melgan: Generative adversarial networks for conditional waveform synthesis". In: *Advances in Neural Information Processing Systems*. 2019, pp. 14881–14892.

[55] Park Kyubyong. *A TensorFlow Implementation of DC-TTS: yet another text-to-speech model*. https://github.com/Kyubyong/dc_tts. 2018.

[56] Sohaib Laraba, Mohammed Brahimi, Joëlle Tilmanne, and Thierry Dutoit. "3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images". In: *Computer Animation and Virtual Worlds* 28.3-4 (2017).

[57] Eva Lasarcyk and Jürgen Trouvain. "Imitating conversational laughter with an articulatory speech synthesizer". In: *Proc. Interdisciplinary Workshop on the Phonetics of Laughter*. 2007.

[58] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[59] Younggun Lee, Azam Rabiee, and Soo-Young Lee. "Emotional End-to-End Neural Speech synthesizer". In: *arXiv preprint arXiv:1711.05447* (2017).

[60] Steven R. Livingstone and Frank A. Russo. "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English". In: *PLOS ONE* 13.5 (May 2018), pp. 1–35. DOI: 10.1371/journal.pone.0196391.

[61] Zhaojie Luo, Tetsuya Takiguchi, and Yasuo Ariki. "Emotional voice conversion using deep neural networks with MCC and F0 features". In: *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*. IEEE. 2016, pp. 1–5.

[62] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[63] Hector P Martinez, Yoshua Bengio, and Georgios N Yannakakis. "Learning deep physiological models of affect". In: *IEEE Computational Intelligence Magazine* 8.2 (2013), pp. 20–33.

[64] Julian McAuley, Rahul Pandey, and Jure Leskovec. "Inferring networks of substitutable and complementary products". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 785–794.

[65] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. 2015, pp. 18–25.

[66] George A Miller. "The magical number seven, plus or minus two: Some limits on our capacity for processing information." In: *Psychological review* 63.2 (1956), p. 81.

[67] Hiroki Mori, Tomohiro Nagata, and Yoshiko Arimoto. "Conversational and Social Laughter Synthesis with WaveNet". In: *Proc. Interspeech 2019*. 2019, pp. 520–523. DOI: 10.21437/Interspeech.2019-2131. URL: http://dx.doi.org/10.21437/Interspeech.2019-2131.

[68] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications". In: *IEICE TRANSACTIONS on Information and Systems* 99.7 (2016), pp. 1877–1884.

[69] Tomohiro Nakatani, Shigeaki Amano, Toshio Irino, Kentaro Ishizuka, and Tadahisa Kondo. "A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments". In: *Speech Communication* 50.3 (2008), pp. 203–214.

[70] Kim Namju and Park Kyubyong. *Speech-to-Text-WaveNet*. https://github.com/buriburisuri/speech-to-text-wavenet. 2016.

[71] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". In: *SSW*. 2016.

[72] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. "Parallel wavenet: Fast high-fidelity speech synthesis". In: *International conference on machine learning*. PMLR. 2018, pp. 3918–3926.

[73] Juan Rafael Orozco-Arroyave, JC Vdsquez-Correa, Florian Hönig, Julián D Arias-Londono, Jesús Francisco Vargas-Bonilla, Sabine Skodda, Jan Rusz, and E Noth. "Towards an automatic monitoring of the neurological state of Parkinson's patients from speech". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 6490–6494.

[74] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[75] Wayland M Parrish. "The concept of "naturalness"". In: *Quarterly Journal of Speech* 37.4 (1951), pp. 448–454.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[77] Hanchuan Peng, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005), pp. 1226–1238.

[78] Gueorgui Pironkov, Stephane Dupont, and Thierry Dutoit. "Multi-task learning for speech recognition: an overview." In:

[79] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. "Context-dependent sentiment analysis in user-generated videos". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1. 2017, pp. 873–883.

[80] Jonathan Posner, James A Russell, and Bradley S. Peterson. "The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology." In: *Development and psychopathology* 17 3 (2005), pp. 715–34.

[81] Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. "A Comparison of Sequence-to-Sequence Models for Speech Recognition." In: *Interspeech*. 2017, pp. 939–943.

[82] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. "Learning to generate reviews and discovering sentiment". In: *arXiv preprint arXiv:1704.01444* (2017).

[83] Tuomo Raitio, Ramya Rasipuram, and Dan Castellani. "Controllable neural text-to-speech synthesis using intuitive prosodic features". In: *arXiv preprint arXiv:2009.06775* (2020).

[84] Ann Ratcliff, Sue Coughlin, and Mark Lehman. "Factors influencing ratings of speech naturalness in augmentative and alternative communication". In: *Augmentative and alternative communication* 18.1 (2002), pp. 11–19.

[85] EH Rothauser. "IEEE recommended practice for speech quality measurements". In: *IEEE Trans. on Audio and Electroacoustics* 17 (1969), pp. 225–246.

[86] James A Russell. "A circumplex model of affect." In: *Journal of personality and social psychology* 39.6 (1980), p. 1161.

[87] Marc Schröder. "Emotional speech synthesis: A review". In: *Seventh European Conference on Speech Communication and Technology*. 2001.

[88] Slava Shechtman and Alex Sorin. "Sequence to sequence neural speech synthesis with prosody modification capabilities". In: *arXiv preprint arXiv:1909.10302* (2019).

[89] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4779–4783.

[90] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron". In: *International Conference on Machine Learning*. 2018, pp. 4693–4702.

[91] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. "Char2wav: End-to-end speech synthesis". In: *ICLR2017 workshop submission* (2017).

[92] Shiva Sundaram and Shrikanth Narayanan. "Automatic acoustic synthesis of human-like laughter". In: *The Journal of the Acoustical Society of America* 121.1 (2007), pp. 527–535.

[93] Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4784–4788.

[94] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. "Voiceloop: Voice fitting and synthesis via a phonological loop". In: *arXiv preprint arXiv:1707.06588* (2017).

[95] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. "A survey on deep transfer learning". In: *International conference on artificial neural networks*. Springer. 2018, pp. 270–279.

[96] Noé Tits. "A Methodology for Controlling the Emotional Expressiveness in Synthetic Speech - a Deep Learning approach". In: *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. 2019, pp. 1–5. DOI: 10.1109/ACIIW.2019.8925241.

[97] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "ASR-based Features for Emotion Recognition: A Transfer Learning Approach". In: *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 48–52. URL: http://aclweb.org/anthology/W18-3307.

[98] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Emotional Speech Datasets for English Speech Synthesis Purpose: A Review". In: *Intelligent Systems and Applications*. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 61–66. ISBN: 978-3-030-29516-5.

[99] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Exploring Transfer Learning for Low Resource Emotional TTS". In: *Intelligent Systems and Applications*. Ed. by Yaxin Bi, Rahul Bhatia, and Supriya Kapoor. Cham: Springer International Publishing, 2020, pp. 52–60. ISBN: 978-3-030-29516-5.

[100] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Neural Speech Synthesis with Style Intensity Interpolation: A Perceptual Analysis". In: *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '20. Cambridge, United Kingdom: Association for Computing Machinery, 2020, pp. 485–487. ISBN: 9781450370578. DOI: 10.1145/3371382.3378297. URL: https://doi.org/10.1145/3371382.3378297.

[101] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "The Theory behind Controllable Expressive Speech Synthesis: A Cross-Disciplinary Approach". In: *Human-Computer Interaction*. IntechOpen, 2019. DOI: 10.5772/intechopen.89849. URL: http://dx.doi.org/10.5772/intechopen.89849.

[102] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "ICE-Talk: an Interface for a Controllable Expressive Talking Machine". In: *Proc. Interspeech 2020*. 2020, pp. 482–483.

[103] Noé Tits, Kevin El Haddad, and Thierry Dutoit. "Laughter Synthesis: Combining Seq2seq Modeling with Transfer Learning". In: *Proc. Interspeech 2020*. 2020, pp. 3401–3405. DOI: 10.21437/Interspeech.2020-1423. URL: http://dx.doi.org/10.21437/Interspeech.2020-1423.

[104] Noé Tits, Fengna Wang, Kevin El Haddad, Vincent Pagel, and Thierry Dutoit. "Visualization and Interpretation of Latent Spaces for Controlling Expressive Speech Synthesis through Audio Analysis". In: *Proc. Interspeech 2019*. 2019, pp. 4475–4479. DOI: 10.21437/Interspeech.2019-1426. URL: http://dx.doi.org/10.21437/Interspeech.2019-1426.

[105] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network". In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 5200–5204.

[106] Jürgen Trouvain. "Phonetic aspects of "speech-laughs"". In: *Oralité et Gestualité: Actes du colloque ORAGE, Aix-en-Provence. Paris: L'Harmattan*. 2001, pp. 634–639.

[107] J. Urbain, H. Çakmak, and T. Dutoit. "Evaluation of HMM-based laughter synthesis". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 7835–7839.

[108] Mohammed Usman, Mohammed Zubair, Mohammad Shiblee, Paul Rodrigues, and Syed Jaffar. "Probabilistic Modeling of Speech in Spectral Domain using Maximum Likelihood Estimation". In: *Symmetry* 10.12 (2018), p. 750.

[109] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit". In: (2017).

[110] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. "Generalized end-to-end loss for speaker verification". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4879–4883.

[111] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. "Tacotron: Towards End-to-End Speech Synthesis". In: *INTERSPEECH*. 2017.

[112] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis". In: *International Conference on Machine Learning*. 2018, pp. 5180–5189.

[113] Oliver Watts, Gustav Eje Henter, Jason Fong, and Cassia Valentini-Botinhao. "Where do the improvements come from in sequence-to-sequence neural TTS?" In: *Proc. 10th ISCA Speech Synthesis Workshop*. 2019, pp. 217–222. DOI: 10.21437/SSW.2019-39. URL: http://dx.doi.org/10.21437/SSW.2019-39.

[114] Oliver Watts, Gustav Eje Henter, Thomas Merritt, Zhizheng Wu, and Simon King. "From HMMs to DNNs: where do the improvements come from?" In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 5505–5509.

[115] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), p. 9.

[116] Zhizheng Wu, Oliver Watts, and Simon King. "Merlin: An open source neural network speech synthesis system". In: *Proc. SSW, Sunnyvale, USA* (2016).

[117] G. N. Yannakakis, R. Cowie, and C. Busso. "The ordinal nature of emotions". In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. Vol. 00. 2017, pp. 248–255. DOI: 10.1109/ACII.2017.8273608. URL: doi.ieeecomputersociety.org/10.1109/ACII.2017.8273608.

[118] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[119] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. "Tensor fusion network for multimodal sentiment analysis". In: *arXiv preprint arXiv:1707.07250* (2017).

[120] Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. "Memory Fusion Network for Multi-view Sequential Learning". In: *arXiv preprint arXiv:1802.00927* (2018).

[121] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech". In: *arXiv preprint arXiv:1904.02882* (2019).

[122] Heiga Zen, Andrew Senior, and Mike Schuster. "Statistical parametric speech synthesis using deep neural networks". In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* IEEE. 2013, pp. 7962–7966.

[123] Heiga Zen, Keiichi Tokuda, and Alan W Black. "Statistical parametric speech synthesis". In: *Speech Communication* 51.11 (2009), pp. 1039–1064.

List of Figures

List of Tables

This thesis was made using a customized version of “hepthesis”.