Analysis of Latent Representations of Neural Text-To-Speech Models for Expressive Audio-Visual Synthesis

Martin Lenglet

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : EEATS - Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)
Spécialité : Signal Image Parole Télécoms
Unité de recherche : Grenoble Images Parole Signal Automatique

Analyse des Représentations Latentes des Modèles de Text-To-Speech Neuronaux pour le Contrôle de la Synthèse Audio-Visuelle Expressive

Analysis of Latent Representations of Neural Text-To-Speech Models for Expressive Audio-Visual Synthesis

Présentée par :

Martin LENGLET

Direction de thèse :

Gérard BAILLY DIRECTEUR DE RECHERCHE, CNRS	Directeur de thèse
Olivier PERROTIN Chargé de recherche, CNRS	Co-directeur de thèse

Rapporteurs :

Marie TAHON
PROFESSEURE DES UNIVERSITES, Université du Mans
Simon KING
FULL PROFESSOR, University of Edinburgh

Thèse soutenue publiquement le **12 décembre 2023**, devant le jury composé de :

Didier SCHWAB, PROFESSEUR DES UNIVERSITES, Université Grenoble Alpes	Président
Gérard BAILLY, DIRECTEUR DE RECHERCHE, CNRS DELEGATION ALPES	Directeur de thèse
Marie TAHON, PROFESSEURE DES UNIVERSITES, Université du Mans	Rapporteure
Simon KING, FULL PROFESSOR, University of Edinburgh	Rapporteur
Gustav EJE HENTER, ASSISTANT PROFESSOR, KTH Royal Institute of Technology	Examinateur

Acknowledgements

This research has received funding from the BPI project Theradia and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). This work was granted access to HPC/IDRIS under the allocation 2021-AD011011542R1 made by GENCI.

Merci à Gérard et Olivier pour votre accompagnement pendant ces trois années. Merci de m'avoir laissé cette chance malgré mon manque d'expérience dans le domaine. Vous m'avez toujours écouté avec beaucoup d'attention, et vous avez su me donner confiance en mes capacités à mener des recherches. Merci également à tous les membres du GIPSA-lab qui ont participé à rendre ces trois années très agréables. Malgré ma réputation de grizzli au fond de sa grotte, je me suis toujours senti bienvenu dans vos bureaux quand j'avais besoin d'un soutien technique ou moral. Merci enfin à tous mes proches pour leur soutien à toute épreuve. Vous êtes tous des exemples qui me poussent à être le meilleur de moi même.

Contents

List of Figures

List of Tables

Table of acronyms and abbreviations

AU	Action Units (Visual Features to control the animation of the virtual avatar)
CMOS	Comparative Mean Opinion Score
CWT	Continuous Wavelet Transform [Vainio et al., 2013]
DTW	Dynamic Time Warping
EoS	End-of-Sequence
E2E	End-to-End
FFT	Feed-Forward Transformer [Ren et al., 2019]
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Models
GST	Global Style Tokens Y. Wang et al., 2018
HMM	Hidden Markov Models
LDA	Linear Discriminant Analysis
LLM	Large-Language-Model
LST	Local Style Tokens Lenglet et al., 2023b
L2S	Letter-To-Sound
MDS	Multi-Dimensional Scaling
MOS	Mean Opinion Score
MSE	Mean Squared Error
NEB	Nadine Eckert-Boulet, French Female Speaker of the LibriVox database
POS	Part Of Speech
RNN	Recurrent Neural Layer
RPT	Rapid Prosody Transcription [Cole & Shattuck-Hufnagel, 2016]
RTF	Real-Time-Factor
TTS	Text-To-Speech

VAE	Variational Auto-Encoder
VAS	Visual Attention Span
XAI	Explainable Artificial Intelligence

Glossary

Attention Lookahead	See Section 2.2.2: 1 or 2 frames of focus of the attention network on one or several characters ahead, followed by a return to the character previously focused.
Characters	In the manuscript, symbols used to write orthographic sequences are interchangeably referred to as characters, letters or orthographic symbols.
Complex Phone	One phone written with several graphemes in the corresponding orthographic transcription.
Embedding	The terms “embedding”, “intermediate representations”, “hidden states” and “latent spaces” are used interchangeably to refer to the intermediation representations computed by neural models between each layer.
Linguistic prosody	Linguistic prosody refers to language-related variations of intonation and rhythm encoded by the text sequence produced.
Paralinguistic prosody	In opposition with linguistic prosody, paralinguistic prosody refers to remaining variations of intonation and rhythm when the linguistic content has been taken into account.
Residual Attention Focus	See Section 2.2.2: the attention pattern characterized by multiple focuses on one character after the initial focus.
Segmental Acoustic Features	Acoustic features specific to phone identity, related to the spectral target to produce to allow perception of this phone.
Supra-Segmental Acoustic Features	Acoustic features whose perception is achieved at a wider scale than isolated phones. These features are relative to prosody.

Introduction

0.1 Background

Even before the disruption of the field by neural networks, voice generation technologies found their applications in various domains. From announcements made through loudspeakers in train stations to instructions provided by GPS devices, speech synthesis allows for an additional mode of interaction between users and computer systems. Voice generation from textual input is a significant accessibility challenge in an increasingly digitalized world, enabling features like automatic website or books reading for visually impaired people [Latif et al., 2020]. Synthetic voices has also gained increasing interest in the medical field, facilitating voice generation for individuals with laryngeal or physio-cognitive disorders [J.-X. Zhang et al., 2021].

Despite recent groundbreaking progress in the naturalness of synthetic voices, these generative technologies face challenges in replicating the richness of natural human interaction. Neural models, which dominate today's synthesis systems, automatically learn to extract linguistic regularities from extensive data corpora, enabling them to generate text sequences never seen before. This statistical learning method has a dual nature: on one hand, it yields plausible and intelligible synthesis, but on the other hand, it tends to erase the pervasive irregularities found in natural speech. These occasional and sporadic variations contribute significantly to the expressive character of human voices. Consequently, their absence quickly leads to disinterest in synthetic voices, which represents a major obstacle to their integration into interactive applications [Potdevin, 2020].

Learning to produce speech from text alone is inherently limited due to the normative nature of orthographic transcriptions. Character sequences only report the linguistic content of the message, but lack the co-verbal aspect that takes an important role in human communication [Streeck & Knapp, 1992]. During an interaction, an important part of the message is actually conveyed by the intonation, which can either nuance or even go against the transmitted linguistic content. Bolinger [1989] distinguishes between two forms of intonation: *emotions* which refer to the mental state of the speaker and therefore are independent of the linguistic content (i.e. paralinguistic content), and *attitudes* that indicate the speaker position toward his/her message. Attitudes are thus characterized by an interaction between the linguistic content and the speaker communicative intent. Paralinguistic content is further impacted by speaker idiosyncrasies: speaker's characteristics like age, education and social background have been shown to impact speech production [Becker, 2014; Foulkes & Docherty, 2006; Resnick, 2012; Stuart-Smith et al., 2014].

Consequently, intonation is not determined solely by the linguistic content. Therefore, neural synthesis frameworks have evolved to take this additional layer of variability into account: textual entries have been augmented with labels which are used to introduce biases into

synthetic models. However, the variability of speech production and perception [Bachorowski, 1999], combined with the entanglement of linguistic and paralinguistic factors make the extensive expressive labeling of corpora very challenging. The main alternative proposed in the literature is the implicit modeling of paralinguistic factors from the audio itself. More specifically, neural architectures were proposed to extract paralinguistic representations from an audio signal, once linguistic and speaker information are already modeled [Min et al., 2021; Skerry-Ryan et al., 2018]. Nevertheless, these unsupervised representations add further intricacy to neural models that are already challenging to interpret. The interpretability of these systems is an issue for the control of these technologies in interactive environments, as a deeper understanding of how neural models encode information is required to design meaningful architectures [Zeiler & Fergus, 2014].

Obviously, gestures also play a role in conveying the co-verbal aspect of communication during natural face-to-face interactions. This is one of the reason why virtual conversational agents have become such an important part of interactive applications for more than 20 years [Cassell, 2000], in areas as diverse as customer services [Bolton et al., 2018] or healthcare [Laranjo et al., 2018]. If integrated successfully, embodied virtual agents introduce a sense of presence [Biocca et al., 2001] which favors the engagement into the interaction [Potdevin, 2020]. The sense of presence is characterized by the importance given by one speaker to the discussion partner in the interaction, and by extension the speaker importance given by the discussion partner in return. Therefore, conveying this sense of presence requires emotional capacities from the virtual agent, both in terms of active listening and expressively adapted speech production [Potdevin, 2020]. This is the promise of the Theradia project [Tarpin-Bernard et al., 2021], which aims to create a virtual agent capable of accompanying patients undergoing cognitive remediation therapy from home. This agent will accompany patients between exercises, giving them feedback tailored to their performance.

0.2 Research Aims

This thesis is part of the Theradia project, and focuses on the generation of verbal and co-verbal behavior in this embodied virtual avatar. The expressive capabilities of this avatar are at the heart of Theradia's desire to offer personalized support, adapted to the patient's emotional state. A number of communication intentions have been highlighted as being important for the avatar to be able to correctly accompany the patient in his exercises. These intentions include (non-exhaustive list): "Comforting", "Committed", "Enthusiastic", "Sorry", etc. These highlighted intentions focus on the transmission of contextual information in parallel with the linguistic content of the message, and thus fall under the definition of attitudes from Bolinger [1989]. Therefore, the explicit control of the expressive capabilities of this avatar, and more specifically the modulation of these attitudes based on the linguistic content, is the end-goal of this research.

To fulfill this goal, several experiments were performed in order to propose an audio-visual expressive generative model from textual content, potentially augmented with attitude-labels. Our main line of research toward this goal is the understanding of internal representations computed by the proposed generative model. Specifically, we propose analytical methods to

probe these representations, in order to exhibit the acoustic features encoded into these latent spaces. We thus provide a data-driven interpretation of the specific task performed by each neural layer of the studied models. This desire to make neural representations interpretable, which contrasts with the concept of models as black boxes, has two main aims:

- Statistical learning performed by neural models constitutes a valuable source of information on natural language, which is hidden within latent representations learned in an unsupervised manner. The proposed probing of encoded information aims to unveil insights about the human speech production mechanisms from low-level phonetic co-variations to high-level phonological organization of sounds, thus bridging the gap between generative speech technology and speech science.
- The identification of features encoded in intermediate layers also enables the design of more careful control mechanisms for neural models. First, the understanding of the specific role of each layer allows us to envision the design of more meaningful subtasks which eases the model training and grants explicit control of some features during inference. Second, we show that highlighting acoustic features through the proposed linear probing provides an explicit post-hoc control of those features, more respectful of the natural covariations learned through statistical training.

Through our work, we aim to provide analysis tools that enable to interpret with greater precision the underlying processes learned by TTS models. This knowledge is essential for a better understanding of the improvements achieved by new models and for designing more effective and meaningful models in the future. For this reason, we place particular importance on the universality of the methods we propose.

0.3 Chapter Overview

Chapter 1 describes the main challenges of the training of neural text-to-speech systems, and further explains the state-of-the-art architectures that are used in the following chapters. A particular focus is given to the specificity of modeling expressivity in these architectures. This chapter also emphasizes the necessary changes to the frameworks used to evaluate these models, now that synthetic voices have reached such high standards in terms of naturalness.

Chapter 2 establishes the baseline synthesis models we designed as the base-implementation for expressive control. Two architectures are chosen for their prevalence in expressive-TTS: Tacotron2 and FastSpeech2.

Chapter 3 explores our proposition of linguistic prosodic control through input data. We propose a new segmentation process for training corpora to better account for the limited capabilities of modeling long-term dependencies in recurrent neural networks. This segmentation in shorter utterances is accompanied by the introduction of inter-utterances linking punctuation marks to convey contextual information between utterances.

Chapter 4 presents our method of linear probing to explore latent representations of TTS models. This exploration reveals some of the underlying learning mechanisms of neural TTS.

Chapter 5 turns the interpretation of internal representations provides by Chapter 4 into explicit control mechanisms for highlighted acoustic features. We also show the benefits of this post-hoc control through learned linear biases compared to explicit control.

Chapter 6 extends the prosodic control established in Chapter 5 to the control of explicit expressive labels. We propose an auxiliary module for this control, which further modulates the expressive bias based on the textual content. Our proposed module emphasizes the benefits of taking the textual content into account when modeling attitudes.

Chapter 7 describes how this work was adapted to the audio-visual synthesis of a conversational agent for the Theradia project.

Chapter 1

Expressive TTS by Biasing the Text-to-Speech Mapping

Contents

The state of the art presented in this section provides an introduction to the key concepts in the field of neural speech synthesis: what are the main models for converting text into speech (or Text-To-Speech, abbreviated TTS in this section)? What are the specific challenges of expressive control? Can the evaluation frameworks follow the groundbreaking performances of latest synthetic models? The goal of this chapter is to introduce the work I performed during my PhD in relation to this literature.

1.1 Neural End-to-End TTS Architectures

Neural TTS models are now ubiquitous in the speech synthesis area. As an extension of the machine learning framework proposed by Hidden Markov Models (HMM)[1], neural models pursue the data-driven approach by extending the number of intermediate representations, called hidden layers. Instead of relying on the careful extraction of features of interest for the designed task, this multiplication of intermediate layers allows neural models to learn sequence-to-sequence mapping from raw large-scale dataset, with minimum supervision of the actual task performed by each neural layer. Indeed, the intermediate representations computed by neural models are mostly structured to maximize the prediction accuracy of the model's output.

More specifically, neural TTS models follow the supervised deep-learning paradigm, which means that models are trained to predict a target output given the corresponding input. Neural models are composed of several stacks of thousands of trainable parameters that compute intermediate differentiable operations. As a result, for any input x, the model is able to return an output prediction $\hat{y} = f(x)$, with f being the functions' composition applied by the neural network on the input x. In order to maximize the prediction accuracy of the model, this prediction $\hat{y}$ should be as close as possible as the target output y for the input x. As a way to deal with memory constraints inherent to large datasets, during the training of neural networks, the error L between the target output y and the prediction $\hat{y}$, called the **loss**, is computed on limited sub-sets of inputs x, called **batches**. Because only differentiable functions are used in the network, the individual contributions of every trainable parameter of the network in the loss L can be computed. This loss gradient indicates how to update the parameters of the network in order to reduce this loss. The training process is then iterative, reducing the loss after each batch of input sequences. This overall training process is usually performed via **back-propagation**, and is the basis of supervised-learning for neural networks[2].

Supervised deep-learning algorithms are inherently limited by the availability of training datasets to tune trainable parameters. Yet neural approaches have outperformed most of classical statistical human-designed predictive models in complex tasks where such datasets are available. TTS is no exception: during the last two decades, rule-based and concatenative synthesis[3] have been mostly replaced by statistical approaches. HMM have showed to potential of statistical learning for voice generation, but statistical approaches could not compete with concatenative synthesis in terms of naturalness until neural models gained increased interest in speech related tasks after 2015. The introduction of neural vocoders like WaveNet [Van den Oord et al., 2016], which were able to generate high quality speech samples from compact acoustic features like mel-spectrograms, marked the first step toward the transition from

[1] For a review of TTS technologies before neural networks, see: P. Taylor [2009], p.422-527.

[2] This process has been refined since its introduction in the 1980s, but only the main ideas are reported in this manuscript for the sake of clarity.

[3] Concatenative synthesis constructs utterances from pre-recorded human speech samples. As a result, the produced synthesis may sound very natural, provided that the recorded corpus covers enough unit combinations. This method is still used in some commercial applications.

HMM to neural architectures. These vocoders paved the way for the subsequent adoption of neural models as Sequence-to-Sequence (S2S) architectures for generating high-quality mel-spectrograms from character/phone inputs. S2S enhances the data-driven approach initiated by HMM, by replacing decision trees by deep trainable hidden representations. Despite the loss of interpretability of such neural models, this shift in paradigm allowed the production of new speech samples that now compete with human recordings in terms of naturalness. This section describes the specificity of neural TTS and the main architectures used in the literature. Two models are given a particular focus for their ubiquity in the field: Tacotron2 [Shen et al., 2018] and FastSpeech2 [Ren et al., 2021].

1.1.1 Choice of Representations

In neural TTS, models are trained to predict speech samples from a sequence of input symbols. In practice, these are two sequences whose elements can be defined in several ways. Input symbols are generally composed of a sequence of characters and/or phones, with the potential addition of punctuation marks or any other type of annotations (e.g., prominence labels [Nenkova et al., 2007; Stephenson et al., 2022], etc.). Each input symbol is encoded into a representation vector, called an embedding, whose value is trained during the learning phase in order to minimize the loss of the model. Phone input is usually preferred in the literature: the phone sequence is an effective characterization of the sequence of acoustic targets to produce, which eases the prediction process of TTS. As will be discussed in Section 1.2, the segmental content itself is not sufficient to predict natural sounding speech samples, since paralinguistic factors are poorly expressed in the textual sequence. Additionally, phonetic choices are already tinged with stylistic [Adda-Decker et al., 1999] and sociodemographic factors [Stuart-Smith et al., 2014], which challenges the adequacy of phonetic input as a description of what has to be said, whatever of how it will be said. As an example, three types of phonological choices in French were highlighted as age- and region-dependent [Brognaux & Avanzi, 2015]:

1. The **deletion of schwas** in monosyllabic grammatical words (“j(e) pense”) and at the initial syllable of polysyllabic words (“il lui a d(e)mandé”)

2. The **liaisons**, which are inserted phonemes between the ending "r", "t", "n" or "s" of a word and the first vowel of the next word. In French, liaisons may be mandatory (“les-z-enfants”), forbidden (“Le train-/ arrive.”) or optional (“il est-(t)-attendu”).

3. The **deletion** of /l/ and /ʁ/ in word-final obstruent-liquid clusters, such as in “pénib(le)” or in the singular personal clitic subject pronouns, such as in “i(l) va”.

As a result, the use of phones as input of TTS models either biases the synthetic voice toward a normative impersonal speech behavior, or requires the prediction of the sociophonetic and stylistic variants to produce. In most TTS applications, a front-end phonetizer is used to compute the Letter-To-Sound (L2S) transcription. The most common implementation of such L2S front-end is a pronunciation lexicon (e.g., CMUDict) where each word is assigned

with a phonetic transcription that represents its standard form, which often omits sociophonetic modulations. Alternative are found through more advanced phonetizers which integrate contextual knowledge to compute pronunciation variations of words and linking sounds between words, either by rules (like eSpeak [Dunn & Vitolins, 2019]) or using separately trained neural networks [Yolchuyeva et al., 2019]. However, using a neural phonetizer only defers the question of the stylistic bias in the prediction of the phonetic sequence.

The use of letters as inputs of neural TTS initially found little success, mainly due to the poor performances reported on L2S conversion [J. Taylor & Richmond, 2019] as an initial step of neural TTS. In opaque languages like French or English, the alignment between letters and the phonetic sequence to produce is far from trivial [Bosse & Valdois, 2009]. Consequently, neural TTS initially lacked a sufficiently robust mechanism to perform the alignment between the input and output sequences. Attention mechanisms were able to alleviate this constraint, as will be discussed in subsection 1.1.3.

As for output speech samples, they are usually encoded into mel-frequency spectrograms. Mel-spectrograms are favored over waveforms for their smoother time-domain variations and their invariance to phase, which makes the loss easier to compute. Mel-spectrograms are computed as logarithmic transformation of the frequency dimension of the short-time Fourier transform magnitude spectrogram. The logarithmic transformation is inspired from human-perception of frequencies [Stevens & Volkmann, 1940]. As such, mel-spectrograms better correlate with perceived differences in speech than the linear-spectrogram. Yet, the loss of the phase information makes spectrograms' inversion to audible waveforms challenging. The Griffin-Lim algorithm [Griffin & Lim, 1984] was initially used to estimate the phase spectrum from the magnitude one. This additional information makes it possible to reconstruct the waveform very quickly. This solution is memory-efficient, but the quality of the predicted audio signal is far from that of the original recordings. More complex architectures called neural vocoders are now used to convert mel-spectrograms predicted by TTS models into audio waveforms. These architectures are further described in subsection 1.1.6.

1.1.2 Corpora

As in any data-driven approach, the quality and the quantity of datasets play a crucial role in the training of neural models. Neural TTS are usually trained on several hours of <text|audio> pairs to achieve the best output quality. The main French TTS datasets available are summed up in Table 1.1. Audiobooks are the most common sources of training data, since they provide dozens of hours of relatively high-quality recordings, along with the associated text. The M-AILABS dataset [Solak, 2019] provides aligned audiobooks recorded with Male and Female voices in English, German, Spanish, Italian, Ukrainian and Russian on top of the French corpus. These free public domain audiobook recordings are taken from LibriVox [Kearns, 2014] and are aligned for use in training speech recognition and speech synthesis models. SynPaFlex [Sini et al., 2018] is an annotated version of the French Female Speaker Nadine Eckert-Boulet (NEB), also seen in M-AILABS.

Table 1.1: Main Publicly Available French TTS Datasets. **AB: Audiobooks**, **PS: Parliament Sessions**, **GU: Generated Utterances** [Le Moine & Obin, 2020]. Years of writing are reported as an indicator of the vocabulary used in corresponding resources.

Dataset	Speaker	Duration	Expressive Labels	Content	Years
M-AILABS [Solak, 2019]	Mixed	190h30m	✗	AB	1884-1964
SynPaFlex [Sini et al., 2018]	Female	87h23m	✓	AB	1884-1964
SIWIS [Honnet et al., 2017]	Female	10h13min	✓	AB + PS	1752-2015
Att-Hack [Le Moine & Obin, 2020]	Mixed	30h	✓	GU	2020

Although audiobooks are valuable resources to train TTS models, several limitations of such corpora need to be emphasized. First, public domain audiobooks are usually quite old: the most recent books from M-AILABS were written in 1964. Therefore, the vocabulary used may not fully cover most recent word usage. The diversity of contexts in which particular words are found in the training corpus maximizes the potential for this word to be correctly pronounced at inference. The absence of words such as the "super", "cluster" "covid" or "chiller" loanwords from English introduces systematic errors in TTS predictions. The combination of more recent corpora like SIWIS [Honnet et al., 2017] mitigates this issue.

Second, audiobooks are recorded in a reading aloud setup in controlled environments, most of the time by professional voice actors. While this setup maximizes the overall audio quality an intelligibility of the recorded speech, reading aloud follows different patterns than spontaneous conversational speech [Bailly & Gouvernayre, 2012; McFarland, 2001]. As a result, synthetic voices trained on audiobooks corpora may sound unnatural in interactive environments. Training voices on conversational speaking styles is an increasing demand of the field [Adigwe & Klabbers, 2022; O'Mahony et al., 2022; Székely et al., 2019], supported by the release of conversational datasets like 100,000 Podcasts from Spotify [Clifton et al., 2020]. If the integration of conversational datasets in TTS training would also be beneficial, to the best of my knowledge, such datasets do not exist in French.

Finally, narrative contents of audiobooks introduce a large variety of contexts that are not fully captured by the text transcriptions: direct speech are reported with '«»' but the actual talking character may be missing. Parenthetical elements may be confused with narrative segments. The punctuation contributes to this disambiguation, but is not sufficient in most cases. Additional annotations are required to further enrich the textual content of audiobooks. The SynPaFlex [Sini et al., 2018] corpus is an attempt to enrich audiobook transcriptions with character, prosody and emotion annotations. Expressive-labeled corpora [Honnet et al., 2017; Le Moine & Obin, 2020] open the route toward supervised-training of paralinguistic features; that will be further discussed in Section 1.2.

1.1.3 Recurrent Neural Layers to predict Temporal Sequences

The first examples of Sequence-to-Sequence (S2S) TTS models fully trained on <text| audio> pairs are Tacotron [Y. Wang et al., 2017] and Char2Wav [Sotelo et al., 2017]. These architec-

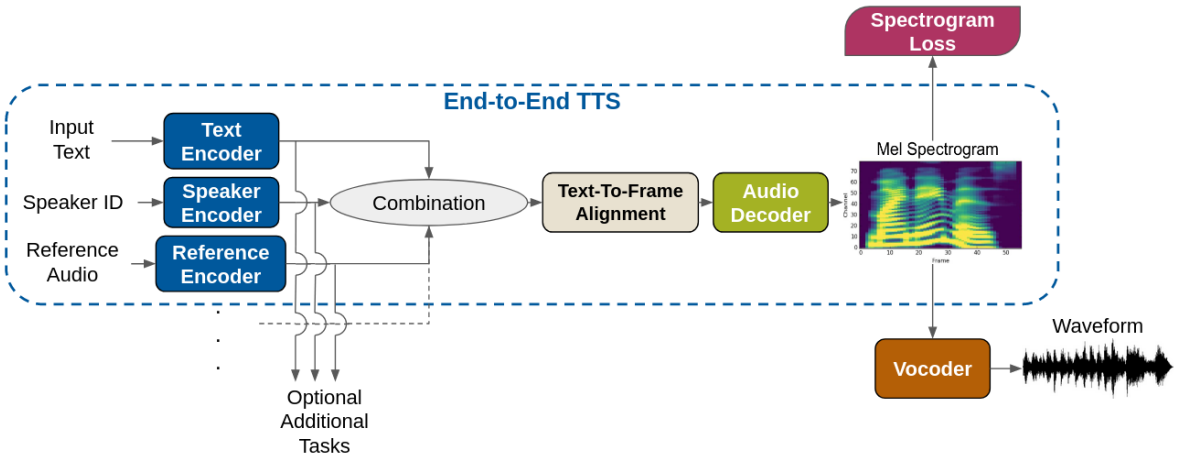


Figure 1.1: Generic Encoder/Decoder Architecture of neural TTS.

tures showed the potential of Recurrent Neural Networks (RNN) to model time dependencies in speech signals. Both models follow the encoder/decoder architecture, illustrated in Fig. 1.1 (in dotted blue line): an encoder converts the input sequence into an abstract representation, which a decoder reads to generate the corresponding sequence of acoustic representations: Char2Wav predicts SampleRNN features [Mehri et al., 2016], while Tacotron is trained to directly generate mel-spectrograms. RNNs are implemented in both the text encoder and the audio decoder: 1) Bidirectional RNNs are used in the encoder in order to contextualize the input sequence (characters for Char2Wav, phones for Tacotron), and 2) a uni-directional RNN introduces causal temporal dependencies in the output sequence. Although alternative structures have been proposed for the encoder and decoder, this architecture is still used in most state-of-the-art systems [Kenter et al., 2019; Ren et al., 2021; Shen et al., 2020; Shen et al., 2018].

1.1.3.1 The Role of Attention in Sequence-to-Sequence Alignment

The alignment of input/output sequences is an essential component of sequence-to-sequence synthesis models. In the case of TTS, mel-spectrogram outputs are on average ten times longer than the input sequences of phones[4]. Duration by phone widely vary between phone-classes: as an example open-vowels tend to be longer than other vowels [O'Shaughnessy, 1981]. Vowel duration also varies with the speaking rate, while most consonants' duration remain stable across speech samples [Nick Campbell, 1992]. Consequently, these non-linearities of the input/output alignment cannot be set by rules. It is therefore necessary to integrate into the model a mechanism to predict the duration associated with each input phone and/or when to switch from the current phone to the next. This mechanism is referred to as duration model. The robustness of the duration model not only ensures the stability of the synthesis (no skipping or repeating of words, no backtracking), but also the naturalness of the voice.

[4]Most common hyperparameters used to compute mel-spectrograms is an hop-size of 256 for an audio sampling frequency of 22.05 kHz, which corresponds to ∼86 Hz or ∼11.6 ms by spectrogram frame. Average duration by vowel measured on our experiments is ∼90 ms.

Statistical HMM TTS models used duration predictors trained from data to solve this alignment problem. This alternative requires prior time-alignments of phones in the training corpus in order to set the targets of the duration predictor during the training phase. Although pre-trained automatic aligners are available for common languages[5], explicit alignment raises 2 main issues:

1. The alignment of silences is unclear. Plosive consonants for example are preceded by a silence caused by the glottal stop of the air-flow. This silence is either merged with the previous word boundary in case of initial consonant, or it is found inside a word and is therefore not supported by any symbol in the text-sequence.

2. There are no explicit alignment rules for orthographic inputs. In opaque languages, several characters are needed to infer the correct pronunciation of the corresponding phoneme [Bosse & Valdois, 2009]. In this case, how is the duration distributed between the graphemes' group?

Provided the establishment of some conventions and the pre-computation of the time alignment on the corpora, duration predictors may also be implemented in neural TTS [Ren et al., 2019; Yu et al., 2020]. These examples are discussed in Section 1.1.4.

In case of absence of explicit alignment, a **recurrent attention network**[6] can be used as an interface between the encoder and the decoder of the model. Following an iterative process, the recurrent attention network computes a fixed-size vector for each new output frame, called the context vector, which enables the decoder to locate itself in the input sequence, and thus to generate the appropriate spectrogram segment [Bahdanau et al., 2014]. This mechanism implements (at least) one Recurrent Neural Network (RNN) layer, which calculates a vector of hidden states s_i for each output time step i. The encoder generates a sequence of vectors $h = \{h_j\}_{j=1}^{L}$, where L is the number of tokens in the sentence to synthesize. For each output frame of index i, the attention network computes a unique attention context vector, denoted c_i. This vector accounts for the importance given to each vector in the input sequence h_j for the generation of the current frame. c_i is calculated by the sum of $\{h_j\}_{j=1}^{L}$, weighted by the alignment weights $\alpha_{i,j}$, following formula. 1.1.

$$c_i = \sum_{j=1}^{L} \alpha_{i,j} h_j \tag{1.1}$$

These alignment weights, denoted α_i for time step i, depend on the hidden states of the attention layer s_i, as well as additional parameters dependent on the chosen attention function. These include context-based attention functions [Chorowski et al., 2015] and position-based attention functions (mostly Gaussian Mixture Models (GMM) [Graves, 2013]). Since the

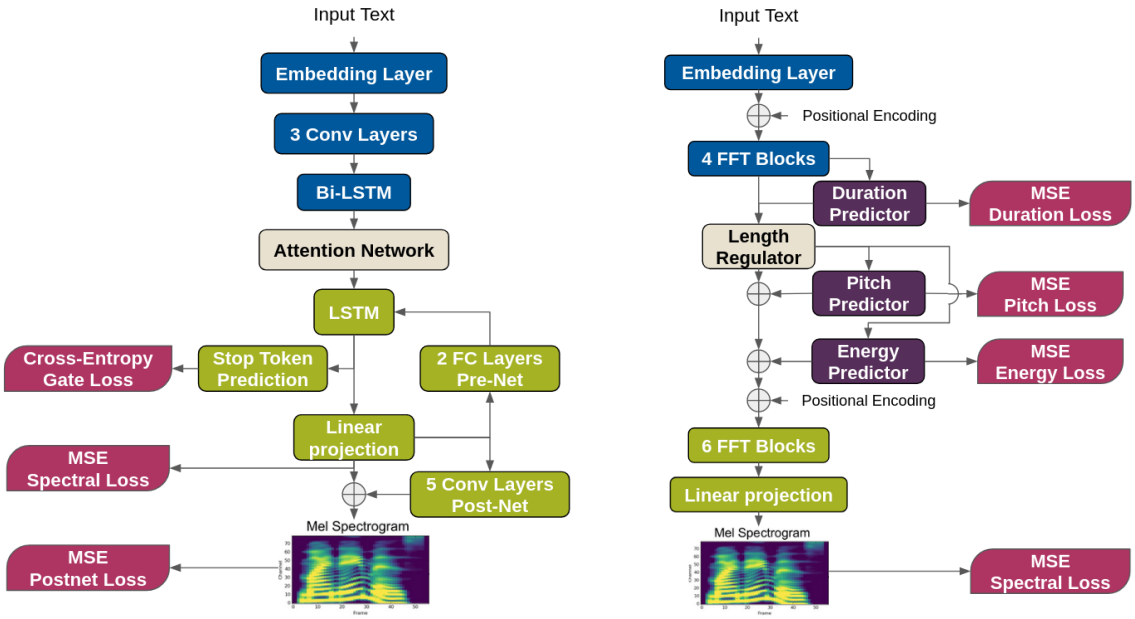
[5] E.g. Montreal Forced Aligner: McAuliffe et al. [2017].

[6] I refer to recurrent attention network in opposition with dot-product attention later proposed by Vaswani et al. [2017]. Dot-product attention does not require recurrent neural networks.

attention weights impact the prediction of the output mel-spectrogram, the attention layer is trained through the spectrogram reconstruction loss.

1.1.3.2 Recurrent Attention-based TTS: Tacotron2

One year after its first version, Tacotron2 [Shen et al., 2018], an enhanced version of Tacotron combined with the neural vocoder WaveNet [Van den Oord et al., 2016], set a new standard by generating a voice that was, at the time, difficult to distinguish from a natural one (average listening test score of 4.53/5, compared with 4.58/5 for a voice recorded under professional conditions). In Tacotron2, illustrated in Fig 1.2a, the encoder (blue) adopts an approach similar to the language models used in other natural language processing applications: the input sequence (characters, phones or both) is transformed into a sequence of trainable embedding vectors. The embedding layer encodes each character (or phone) in the database in a multi-dimensional latent space. These embeddings then pass through three layers of Convolutional Neural Networks (CNN) [Kalchbrenner et al., 2014] of limited span (two characters to the right and left per layer), which introduce a form of local context into the embeddings. The output of these CNN finally passes through a bidirectional Long-Short-Term Memory (bi-LSTM) unit [Hochreiter & Schmidhuber, 1997; Schuster & Paliwal, 1997], which introduces a broader contextualization, for example by coloring embeddings from the beginning of a sentence with the presence of a question mark at the end of the sentence [Stephenson et al., 2020].



(a) Recurrent-TTS: Tacotron2 [Shen et al., 2018]

(b) Transformer-Based TTS: FastSpeech2 [Ren et al., 2021]

Figure 1.2: Two examples of the main types of architectures in neural TTS. **LSTM:** Long-Short-Term Memory, **FC:** Fully Connected, **MSE:** Mean Squared Error, **FFT:** Feed-Forward Transformer.

The output of the encoder is then processed by the context-based attention [Chorowski et al., 2015], which contributes to the monotony of the alignment path (avoiding skipping and rollbacks). The decoding process (green in Fig 1.2a) is autoregressive: for each decoding step, a new context vector is computed from the output of the encoder and from the previously generated spectrogram frames (re-introduced through a bottleneck pre-net). The LSTM[7] layer uses this context vector to predict the next spectrogram frame. In the meantime, the hidden state computed by the LSTM is used by the Stop Token Predictor to predict the end of the generative process. Finally, once a first version of the spectrogram has been predicted, the postnet computes a residual which is added to this spectrogram. This postnet implements five 1-D convolutional layers in time domain, mainly to smoothen the predicted spectrogram coefficients. The entire Tacotron2 model is trained with three losses, following the formula 1.2 (magenta in Fig 1.2a):

$$L_{TC} = L_S + L_{PS} + L_G \tag{1.2}$$

with L_{TC} the total loss of Tacotron2, L_S the MSE spectral loss, L_{PS} the MSE spectral loss after the Postnet and L_G the cross-entropy Gate Loss.

1. The **Spectral Loss** (resp. **Postnet Loss**) computes the error between the ground-truth spectrogram and the predicted spectrogram before (resp. after) the postnet. This dual evaluation of the spectral loss seemingly helps reaching convergence [Shen et al., 2018] during training. The mean squared error is selected as the loss function because it assigns equal importance to all frequency bands computed in the mel scale spectrogram.

2. The **Gate Loss** computes the error of prediction of end-of-sequence. During the training phase, the model is stopped once the output sequence length matches the ground-truth target (following the teacher-forcing training setup). But at inference, given the decoding process is autoregressive, the length of the output sequence is not known in advance. Instead, at each decoding step, the Stop Token Predictor, which is a 1-D linear projection, computes the probability of ending the sequence. During training, this probability is set to 0 during the decoding process, expect for the last frame which should predict 1. This target is compared to the output of the stop token predictor via cross-entropy to compute the Gate Loss.

The recurrent TTS framework embodied by Tacotron and Tacotron2 has been built on since the original introduction of this model. Notably, Tacotron-based architectures have been widely used in the expressive TTS field: Table 1.2 summarizes a selection of recent TTS models based on the Tacotron architecture. The versatility of the recurrent architecture associated to the attention interface, which is able to learn the S2S alignment in an unsupervised manner, makes Tacotron-based models one of the main deep TTS paradigms [Triantafyllopoulos et al., 2023].

[7]The LSTM layer is a type of RNN which implements a long-term memory unit to carry information through longer sequences [Hochreiter & Schmidhuber, 1997].

Table 1.2: Main Neural TTS based on Tacotron or Tacotron2 architectures. **"emb": embedding**, **"ctrl": control**. ↑ and ↓ indicate increase and decrease respectively.

Reference	Model Name	Main Modifications	Main Results
He et al. [2019]	Monotonic Attention	Monotonic attention path	Avoid collapse and repetitions
Skerry-Ryan et al. [2018]	Reference Encoder	Implicit prosody emb	↑ Prosodic transfer
Klimkov et al. [2019]	✗	Phone-Level prosodic emb	↑ Prosodic transfer
Y.-J. Zhang et al. [2019]	VAE Reference Encoder	Projects prosody emb on VAE	↑ Disentanglement
Y. Wang et al. [2018]	GST	Disentanglement of reference emb	↑ Disentanglement
Raitio et al. [2020]	Explicit Prosodic Encoder	Utterance-Wise Encoder	Ctrl of F0, energy, duration, spectral tilt
Mohan et al. [2021]	Ctrl-P	Explicit phone prosodic emb	Ctrl of F0, energy, duration
R. Liu et al. [2021]	✗	Style Reconstruction Loss	↑ Style transfer
Valle et al. [2020]	Mellotron	Explicit prediction of Pitch & Rhythm	Variety of style generation
Shen et al. [2020]	Non-Attentive Tacotron	Replaces attention by duration predictor	↓ Computation time, ctrl speaking rate
Hussen Abdelaziz et al. [2021]	AV-Tacotron2	Predicts facial features from text	Performances ≈ GT extracted features

1.1.4 Emergence of Self-Attention

1.1.4.1 Limits of Recurrent Networks

Despite the convincing results of Tacotron-based TTS, the use of RNNs raises several concerns. Like LSTMs, whose schematics are shown in Fig. 1.3, a RNN computes hidden representations of each element of a sequence as a function of all previous elements in the sequence (and of all subsequent elements in the case of a bi-directional RNN [Schuster & Paliwal, 1997]). The t^{th} element of the sequence x , named x_t in Fig. 1.3, is assigned an output h_t, which depends on the output h_{t-1} and on the memory state c_{t-1} of the previous step. The output h_T thus encodes a causal representation of $x_{t<T}$. This method introduces a form of long-term context, but it is sequential and therefore imposes long computation times. Moreover, training an RNN on long-term dependencies can be tedious because of vanishing gradients [Hochreiter et al., 2001].

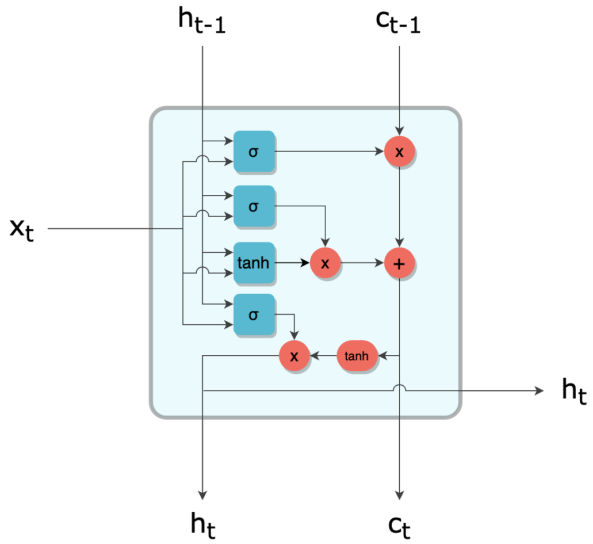


Figure 1.3: Architecture of an LSTM unit [Hochreiter & Schmidhuber, 1997]. Source: https://blog.octo.com/les-reseaux-de-neurones-recurrents-des-rnn-simples-aux-lstm/

In order to mitigate these limitations, Li et al. [2019] proposed to replace RNN layers by self-attention layers [Vaswani et al., 2017] in their model called *Transformer-TTS*. Self-attention consists in the computation of contextualized embeddings by parallel linear projections and dot matrices-products (see Fig. 1.4). Each input token embedding $(x_{i,1}...x_{i,d_{model}})$ – where d_{model} is the dimension of these embeddings and i the position in the sequence – is projected into three distinct spaces: the "Queries" Q, the "Keys" K and the "Values" V. The linear projections are independent of the position of the embedding in the input sequence, their coefficients are learned during training, and their dimension d_k can be different from d_{model}. The product of the Q matrix and K^T gives the E matrix of attention scores. Vaswani et al. [2017] proposes a normalization of these attention scores by dividing by the square-root of d_k, the dimension of the projections on K and Q. By computing the softmax function per row of this matrix, each input embedding is given a probability distribution associated with all the elements in the sequence (itself included). The weighted sum of the rows of the "Values" V by these probability distributions finally provides an in-context representation of the input. Multiple intra-sequence dependencies can be captured thanks to several attention-heads. In this case, each attention head can be specialized in one type of context (adjacent phonemes, long-term context, punctuation, etc.).

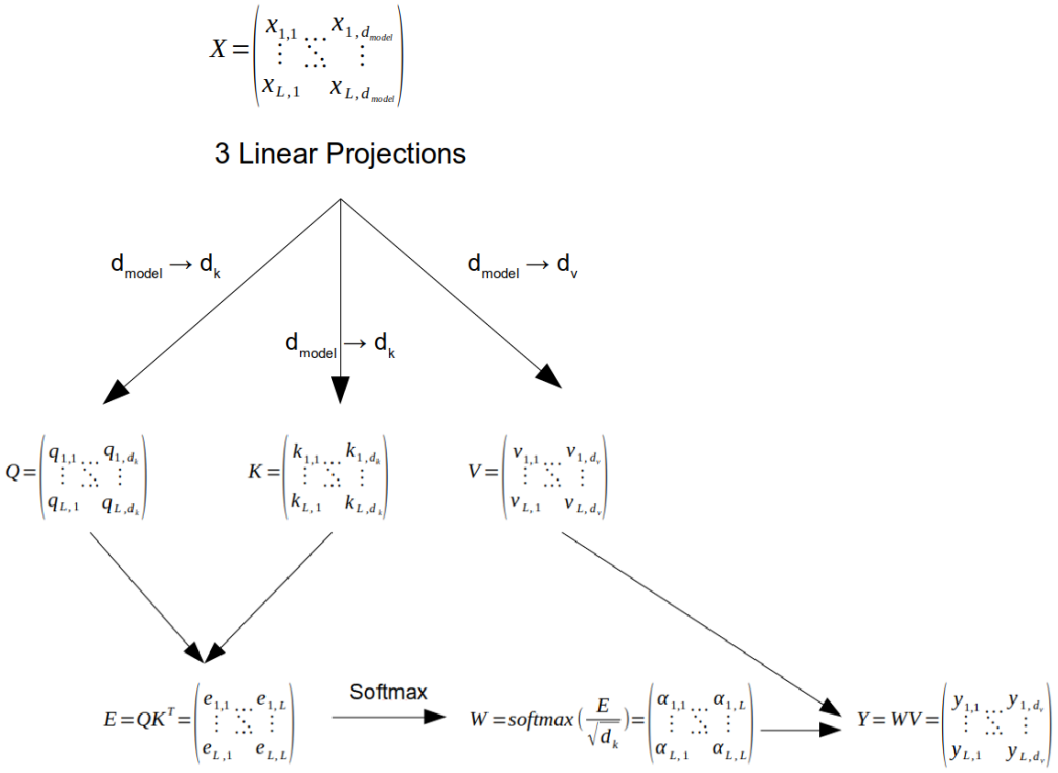


Figure 1.4: Computation Process of Self-Attention Layers.

The self-attention mechanism provides two major benefits compared to RNN:

1. The contextualization process is now parallel instead of sequential. As a result, all the sequence is computed at once in the encoder during training and inference, which drastically reduces the computation time (Transformer-TTS speeds up the training 4.25 times compared to Tacotron2 [Li et al., 2019]). Note that the Transformer-TTS decoder remains autoregressive.

2. The dependency between two elements of the input sequence is directly computed in the matrix E for all pairs of elements in the sequence, regardless of their distance from each other. Doing so, the self-attention mechanism gets rid of the risk of memory loss between distant elements of the sequence.

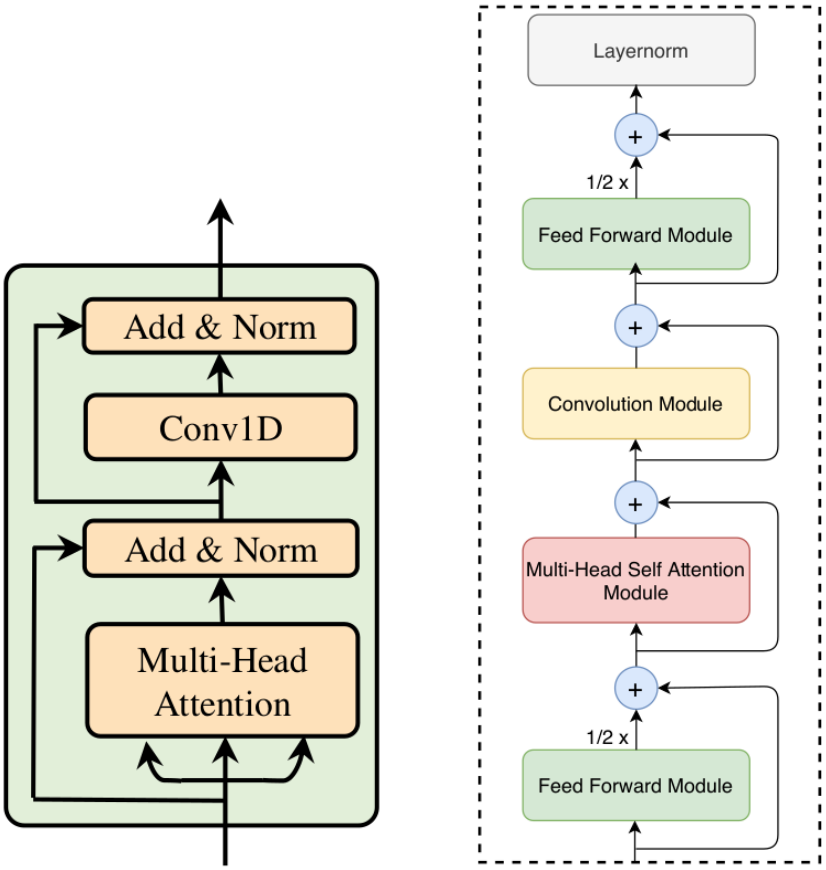
In Transformer-TTS [Li et al., 2019], the encoder and the decoder are implemented by stacks of multi-head self-attention layers followed by Fully-Connected Networks (FC). The recurrent attention is replaced by a dot-product cross-attention. The authors showed that self-attention layers were able to replace RNN and CNN of Tacotron-based architectures, with equivalent performances in terms of naturalness. Nevertheless, to achieve performance similar to that of the Tacotron2, Transformer networks stack up a significant number of self-attention layers (at least 6 to replace one LSTM layer). Transformer models are therefore more complex and more difficult to interpret. The lack of insight into these complex models can impede progress in designing more interpretable and effective architectures.

1.1.4.2 Transformer-Based TTS: FastSpeech2

FastSpeech [Ren et al., 2019] can be seen as a extension of the Transformer-TTS [Li et al., 2019]. Self-Attention Layers are turned into Feed-Forward Transformer (FFT) blocks, illustrated in Fig 1.5a. The recurrent attention mechanism previously implemented in recurrent attention-based TTS to compute the text-to-frame alignment at the interface between the encoder and the decoder is replaced by a explicit duration predictor, which expands the input sequence to match the length of the output spectrogram to predict. The decoding process is thus informed of the entire sequence to produce and process this sequence in parallel, which further increases the inference and training speed. This explicit duration predictor extracts the alignment from pre-computed unsupervised attention paths learned by a pre-trained autoregressive model[8]. The explicit training of this duration predictor ensures the robustness of the alignment, and enables the control of the speaking rate at inference. However, training this model required the pre-training of an attention-based TTS.

FastSpeech2 [Ren et al., 2021], illustrated in Fig 1.2b, alleviates this issue by training the duration predictor on pre-processed time alignments of the training corpus. Additionally, FastSpeech2 implements two explicit prosodic predictors for fundamental frequency and energy respectively. Authors showed that the explicit modeling of prosodic features improved

[8]Ren et al. [2019] originally used the attention paths from a pre-trained Transformer-TTS, but Tacotron-based models would work similarly.



(a) Feed-Forward Transformer (FFT) Block used in FastSpeech and FastSpeech2. Source: [Ren et al., 2019].

(b) Conformer Block used as alternative to FFT: Source: [Gulati et al., 2020].

Figure 1.5: Standard Transformer-Blocks used in neural models.

listeners' opinion on the synthetic voice quality, while allowing the explicit control of these features at inference. Fundamental frequency and energy are predicted for each spectrogram frame. Fundamental contour is predicted from the inverse Continuous Wavelet Transform (CWT) [Vainio et al., 2013] and energy is computed as the norm of the amplitude spectrogram frame. These three explicit prosodic losses (duration, fundamental frequency and energy) are added to the mean absolute spectral loss which is used to train the model, following the formula 1.3.

$$L_{FS} = L_S + L_{dur} + L_p + L_e \tag{1.3}$$

with L_{FS} the total loss of FastSpeech2, L_S the spectrum MAE loss, L_{dur} the MSE duration loss, L_p the MSE pitch loss on CWT and L_e the MSE energy loss.

Despite the additional constraint of the required pre-processing of duration, fundamental frequency and energy prior to training, FastSpeech2 is particularly suited for real-life applica-

tions: the robustness of the duration predictor, combined with the controllability provided by explicit prosodic predictors ensures the overall quality of the produced synthesis. The parallel decoding process endows FastSpeech2 with a very competitive inference speed[9].

The performances of FastSpeech-based TTS are further enhanced by the replacement of Feed-Forward Transformer (FFT) layers (Fig. 1.5a) by Conformer layers [Gulati et al., 2020] (Fig. 1.5b). Similarly to FFT, Conformer combines the benefits of the multi-head self-attention to model global context with the CNN to learn local interactions in the temporal sequences. However, in the Conformer architecture, the convolution module is surrounded by two gated linear units as introduced by Dauphin et al. [2017]. This configuration creates an information bottleneck, which the authors hypothesize that it enables the model to concentrate on the most significant features embedded in the data for computing local dependencies. This replacement has been shown to benefit to TTS performances [Guo et al., 2021; Xu et al., 2023].

These performances established FastSpeech2 as one of the leading architectures among neural TTS, as emphasized by the recurring use of this model in the literature (see Table 1.3).

Table 1.3: Main Neural TTS based on FastSpeech or FastSpeech2 architectures. **"emb": embedding**, **"ctrl": control**. ↑ and ↓ indicate increase and decrease respectively.

Reference	Model Name	Main Modifications	Main Results
S. Lei et al. [2022]	Hierarchical Context Encoder	Phrases & sentences emb from LLM	↑ naturalness
Łańcucki [2021]	FastPitch	F0 prediction at phone-level	Ctrl F0, ↑ naturalness
M. Kim et al. [2021]	Style-Tag-TTS	Linguistic emb space	Ctrl with Free style tag
Min et al. [2021]	StyleSpeech	Style adaptive scaling and shifting	↑ naturalness
Chien et al. [2021]	✗	Pre-trained speaker emb	↑ Voice conversion
Guo et al. [2021]	Conformer-TTS	Replacement of Transformer by Conformer	↓ MCD compared to FastSpeech

1.1.5 From deterministic to probabilistic speech modeling

Tacotron- and FastSpeech-based models stand as references in the TTS field. These two architectures have proved their performances during the latest Blizzard Challenge in 2023 [Perrotin et al., 2023]: among the 18 participants of the Hub-Task, five acoustic models were based on the recurrent layers following the Tacotron2 architecture, and six models on the FastSpeech2 architecture[10] (including the best rated system for the quality assessment of the Hub-Task: MuLanTTS [Xu et al., 2023]). However the architecture of both these models is deterministic: at inference, the output spectrogram is unambiguous predicted from the input sequence of characters or phones. This determinism goes against the one-to-many nature of the TTS framework: the text sequence alone cannot assess unambiguously the acoustic sequence that would be produced by a human speaker, as will be further discussed in Section 1.2. The probabilistic dimension of speech has been absent in the previously discussed architectures but has gained increasing attention in recent years within the field.

[9] Real-Time-Factor is 1.95×10^{-2}, 50 times faster than Transformer-TTS.

[10] Note that three of these six participants who used FastSpeech2-architectures used Conformer layers [Gulati et al., 2020] in their acoustic model.

Variational Auto-Encoders (VAE) [Kingma & Welling, 2013] have been proposed to model the inherent variability of speech representations. VAEs shift the auto-encoder paradigm by introducing the prediction of a probability distribution over a reduced latent space instead of encoding inputs directly as coordinates. Provided the regularization of this reduced space by additional constraints such as the KL-Divergence, the prediction of a distribution allows sampling from this latent space at inference, generating various plausible examples instead of one deterministic prediction. J. Kim et al. [2021] adapted this concept for TTS in their *Variational Inference for end-to-end Text-to-Speech* (VITS). This model still follows the encoder/decoder architecture, but with some adjustments: 1) the model is fully end-to-end, as it predicts waveforms directly from text, thus the decoder refers to the prediction of the waveform from VAE latent features. 2) The encoder computes the prior distribution on the VAE space from the input sequence of character or phones and the learned alignment. 3) The alignment is learned automatically through the maximization of likelihood of the data parametrized with a normalizing flow. This conditional generative architecture maximizes the TTS potential to generate a wide variability of intonations and rhythms[11]. Thus, VAE-based TTS have potential for modeling expressive speech [Shirahata et al., 2023].

1.1.6 Neural Vocoders

As illustrated in Fig. 1.1, a consensus has been reached on the use of mel-spectrograms as speech representations to predict from textual inputs. However, a mel-spectrogram is a highly compressed representation of the original audio signal, in which: 1) the frequency dimension is downsampled compared to a full spectrogram (usually 80 mel-frequency bins compared to 1024 in a full spectrogram amplitude using standard hyper-parameters); 2) the phase is missing[12]. Predicting the waveform from the mel-spectrogram is therefore not a trivial process. That is why most TTS models are usually associated with a vocoder, which performs the conversion from the mel-spectrogram predicted by the TTS model into an audible audio signal. As vocoders are also deep learning networks, they are called neural vocoders (see [Tan, 2023] for a recent extensive review) . The most common are WaveNet [Van den Oord et al., 2016], WaveRNN [Kalchbrenner et al., 2018], Waveglow [Prenger et al., 2019], LPCNet [Valin & Skoglund, 2019], Parallel WaveGAN [Yamamoto et al., 2020], HiFi-GAN [Kong et al., 2020] and more recently diffusion-based vocoders [Koizumi et al., 2023].

In practice, neural vocoders play a minor role in the rendition of prosody: the mel-spectrogram predicted by the S2S model already encodes the phrasing and the intonation. The performances of the neural vocoders only impact the audio fidelity of the reconstructed waveform. For this reason, neural vocoders are not the main focus of this thesis. However, in order to make an informed decision of which vocoder to use, we relied on the benchmark computed by Govalkar et al. [2019]. In this study, authors compared the neural vocoders commonly used in the literature in terms of re-synthesis, i.e. the spectrogram is extracted from the original recording, then converted back into an audio signal by the various vocoders. Note

[11] This potential is validated by the use of VAE-based acoustic models for six participants of the Blizzard Challenge.

[12] A common mel-spectrogram representation thus applies a compression factor of $2048/80 = 25$.

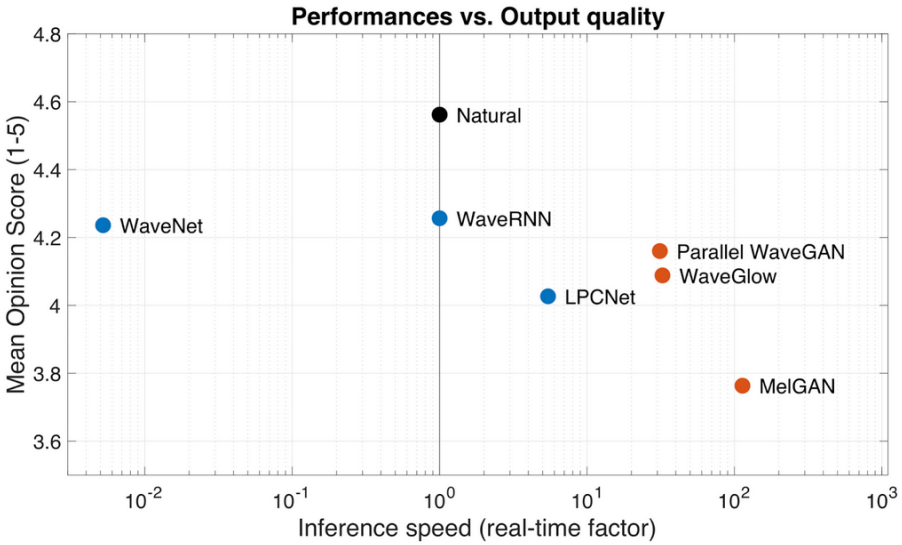


Figure 1.6: Comparison of performances between main neural vocoders. Autoregressive models (resp. parallel) are indicated in blue (resp. orange). Source: [Perrotin, 2021]

however that this comparison focuses on the ideal case of analysis-resynthesis, without taking into account the possible perturbations of the spectrum when predicted by a TTS model. Fig. 1.6 summarizes the compilation of the results of this study with twenty others studies on neural vocoders. Perrotin et al. [2021] compared neural vocoders operating outside their typical operating range and confirmed most of this ranking.

Vocoders can be divided into 2 categories: (1) Autoregressive vocoders, in blue, generate the waveform one sample at a time, taking into account all previously generated samples; (2) Parallel vocoders, in orange, generate the entire waveform from the complete spectrogram. Because of this difference, autoregressive vocoders are generally slower than parallel vocoders, but achieve better signal quality at the output. The choice of vocoder is therefore a compromise between the desired audio quality, and the acceptable inference time for the desired application. It's worth noting, however, that the audio quality of parallel vocoders has been steadily improving in recent years, so they are expected to combine the best of both categories in the short to medium term. As an example, in the Blizzard Challenge 2023 [Perrotin et al., 2023], 16 out of the 18 participants used a parallel vocoder (10 HiFi-GAN [Kong et al., 2020], 3 BigVGAN [S.-g. Lee et al., 2023], 2 Waveglow [Prenger et al., 2019] and 1 StyleMelGAN [Mustafa et al., 2021])[13].

Neural vocoders have mainly been used in TTS following a two-stage generation pipeline (S2S-TTS + Vocoder), replacing signal-based vocoders such as Harmonic+Noise models [Stylianou, 2001] or STRAIGHT [Kawahara et al., 1999]. In practice, when two models are cascaded, they can be trained separately, potentially on different corpora. In the case of TTS and vocoders, this distinction is valuable, since vocoder training datasets do not require the corresponding

[13]The remaining 2 participants used FastDiff [Huang et al., 2022] and a joint acoustic model and vocoder training.

text alignments. As such, vocoders can be trained separately on wider audio-only corpora covering large range of variation of phonation and articulation. They are widely considered as universal, i.e. speaker- and language-independent. This two-stage pipeline then enables the same universal vocoder to be combined with any newly trained TTS in order to generate speech samples.

However, audio quality ratings by human listeners tend to favor the audio samples generated by vocoders that have been fine-tuned on mel-spectrograms predicted by the TTS model for targeted languages and speakers [Kong et al., 2020]. This result indicates that fine-tuning the vocoder on predicted mel-spectrograms reduces the perceived spectral error which remains after the TTS training. The training of joint text-to-waveform models is a step further to this direction [Lim et al., 2022; Miao et al., 2021; Ren et al., 2021]. This procedure maximizes the perceived audio quality of synthetic speech, at the expense of the intermediate computation of human-interpretable representations like mel-spectrograms. This joint training makes the TTS comparison with other models harder, since changes in performances cannot be solely awarded to the modifications of the acoustic model. Therefore, joint training is most relevant once TTS and vocoders architectures are fully settled. Thus, joint training was not explored in this thesis.

1.2 Expressive Synthesis

Despite the progress in naturalness of synthetic voices reported for neural TTS models in Section 1.1, expressive voice control remains a major challenge in the field. Indeed, speech generation from text is a one-to-many problem: text alone is not enough to determine uniquely the prosodic and acoustic structure of the sentence to be pronounced. The speaker's characteristics (age, gender, social background, etc.) [Becker, 2014; Foulkes & Docherty, 2006; Resnick, 2012; Stuart-Smith et al., 2014], his/her physical, mental and emotional state as well as his/her communication intent are numerous factors that have a strong effect on the phonological and phonetic structure of the utterance, in particular its intonation [Cruttenden, 1997; Wichmann, 2000].

While identifying and characterizing speakers is a relatively simple task, and has already been done on most corpora available, unambiguously determining the speaker's communicative intention and/or mental state is far more difficult. The influence of the paralinguistic context (intent of communication, mental state, etc...) on the speech production, referred to as **style** in the literature, is loosely defined. Speaking styles are expressed differently by different speakers, and may be perceived differently by different listeners [Bachorowski, 1999]. This challenge in the interpretation of paralinguistic cues also comes from the entanglement between intentional intonations given by the speaker to convey additional context to the semantic content of the text, and the speaker underlying knowledge and understanding of his/her role in the communication process [Brown et al., 1980].

This section presents the notion of expressive style in neural TTS. This term will be first defined in subsection 1.2.1 in the scope of the contributions presented in the following chapters

of this manuscript. The adaptations of TTS to take this additional layer of variability into account will be further described in subsection 1.2.2. A particular focus will be given to extensive evaluation of stylistic representations in subsection 1.2.3.

1.2.1 Various Layers of Expressive Contributions

In speech production, acoustic realizations are the combinations of numerous contributions, illustrated in Fig 1.7. The cognitive processes in the speaker that are influenced by these factors are beyond the scope of the presented manuscript. Thus I will only discuss the effects of the most prominent constraints to speech production, with regards to the TTS framework.

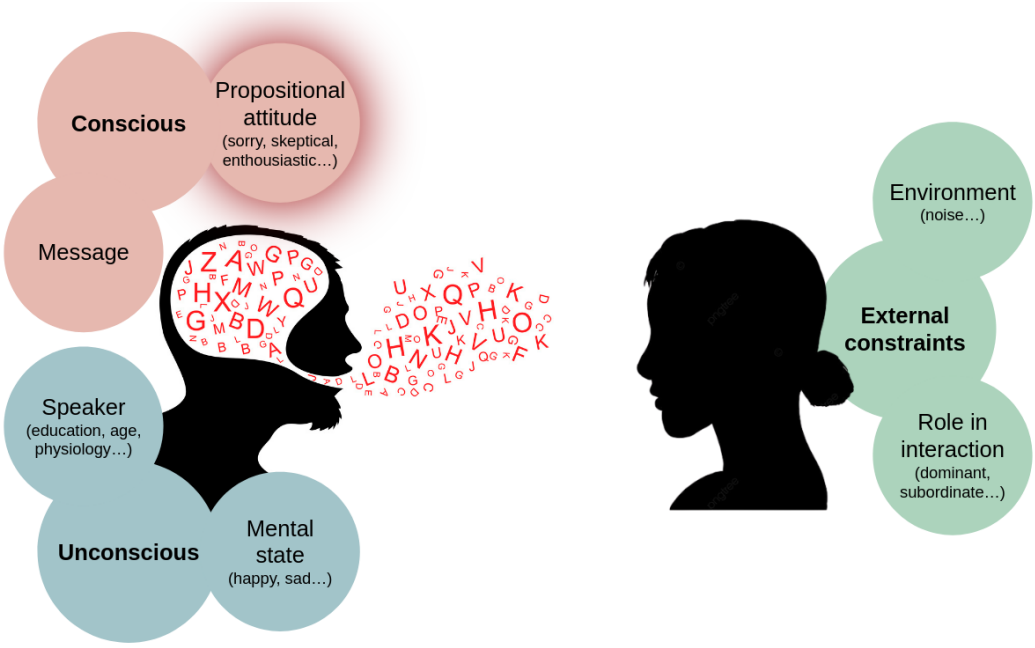


Figure 1.7: Simplified View of the Multiple Contributions to Speech Production

1.2.1.1 Definition of Attitudes

First and foremost, the actual message transmitted by the speaker imposes the sequence of phones to be produced in order to make the message intelligible to the listener. Moreover, intonation and rhythm are, to a certain extent, constrained by the phonological organization of the language [Ladd, 2008]. In some languages, stress may be distinctive for (at least certain) phones, which makes the correct intonation a requirement for phone intelligibility (see [Maeda, 1976] for American English). In other languages – as in French –, stress is not distinctive at phone-level but contributes to the naturalness at word and utterance-level. As an example, accentuation of the final syllable of a word may indicate a word demarcation instead of a deliberate prominence ("bordures" VS "bords durs"[14]). As a whole, this encoding of textual

[14]Example from [Vaissière, 2002]

content and punctuation by intonation and rhythm, referred to as **linguistic prosody**, should be captured by statistical learning from textual inputs through prosodic regularities found in the training corpus.

Conversely, **paralinguistic prosody** requires specific considerations that are discussed below. Obviously, speaker variability plays a role in the speech production. On top of the sociodemographic factors already discussed, *attitudes* may affect intonation. Bolinger [1989] distinguishes between emotions ("how you feel when you say") and attitudes ("how you feel about what you say") that more intricately depend on the message. This distinction is further discussed by Wichmann [2000] who identified two types of attitude[15]: 1) **attitudinal intonations** which are related to the speaker interpretation of his/her role in the communication process, and 2) **propositional attitudes** which reflect the speaker position with regards to what he/she is talking about. Thus, propositional attitudes convey contextual information that the speaker deliberately transmits to the listener for communicative purposes.

1.2.1.2 Application to TTS

The literature on expressive control of TTS generally refers to the control of paralinguistic prosody as **style control**, without distinction between emotions, attitudinal intonations and propositional attitudes. Without further specification, a confusion remains as to the exact definition of the expected scope of such control. As an example, the corpus Att-Hack [Le Moine & Obin, 2020] is specifically recorded to analyze attitudinal intonations, which the authors also call social attitudes. The confusion between emotions and propositional attitudes is further induced by the choice of labels commonly used to describe propositional attitudes, which characterize the planned communicative intent by the emotion induced to the listener ("comforting", "pleading", etc.). Moreover, attitudes are usually inferred by the listener but are used to describe the position of the speaker toward his/her message [Bolinger, 1989].

In the scope of TTS applications, the modeling of these styles is a key factor to favor engagement in the interaction [Potdevin, 2020]. Nonetheless, engaging with a synthetic voice is inherently limited when compared to human interactions, as the synthetic speaker's communicative intentions are primarily directed towards achieving its assigned objectives (encouraging longer interactions for social bots, responding as efficiently as possible to a request for voice assistants, etc.). Thus, we believe that the modeling of propositional attitudes should be given a primary focus in expressive TTS.

However, the identification of the individual impact of each type of paralinguistic contribution to the prosody is a challenging task, as the three presented layers may interact [Brown et al., 1980]. As an example, the absence of low endpoint in the pitch contour may be perceived as a polite indication of the speaker's willingness to keep the speaking-turn, or may be seen as uncertain[16]. Additionally, some attitudes may be conveyed by the mismatch between the

[15] Wichmann [2000] also identified emotions as a contributing factor of speech production, which she referred to as *expressive intonations*.

[16] Example from [Wichmann, 2000]

content and the intonation [Knowles, 1987]. The understanding of such attitudes is therefore harder since the recognition of such mismatch requires the prior identification of the normal association between intonation and content. The entanglement between the described layers of expressiveness is therefore a challenge to the expressive annotation of corpora.

External environmental conditions may also impact speaker intonation. As an example, speakers adapt speech production to increase intelligibility in adverse conditions. This is known as the *Lombard effect*, extensively studied in the literature [Hazan & Baker, 2011; Junqua, 1996; Krause & Braida, 2004]. The specific challenge of this adaptation is the consideration of an auditory feedback in the generation process. Such an example of a *listening* TTS is the Lombard-TTS architecture [Novitasari et al., 2022]. In the following, expressive control is nevertheless discussed in quiet environments.

1.2.2 Attitude Modeling in TTS: Combination of Biases

The main issue with the modeling of expressive styles in TTS is the expressive tagging of corpora. As explained in Section 1.1, the training of neural TTS follows a supervised-learning framework: the model is trained to predict the target speech given the corresponding textual input. We have already mentioned how this textual input may not be sufficient to explain the produced speech. Therefore, the most direct way to augment this input with the required nuances is the introduction of additional inputs to the model, in a way similar to the "Speaker ID" in Fig.1.1. This enhanced training setup thus requires access to expressive-labeled corpora. But, as explained in the previous section, the entanglement of expressive layers limits the extensive labeling of recorded speech. Consequently, two approaches co-exist in order to account for the accessibility of expressive labels: supervised and unsupervised style modeling.

1.2.2.1 Supervised Style Modeling with Labels

With access to style labels, the supervised learning of expressive contribution is equivalent to learning the mapping between these labels and the corresponding acoustic modulations. This training setup is even more simplified in parallel datasets, in which only the expressive label (attitude, speaker, etc.) varies between recordings. As an example, speakers are relatively easy to identify, if not already labeled in most available recordings. For this reason, speakers are mostly modeled explicitly as additional inputs of neural TTS [M. Chen et al., 2020; Skerry-Ryan et al., 2018; Y. Zhang et al., 2019]. More specifically, a set of trainable speaker embeddings are learned alongside the phone or character embeddings. This speaker embedding is combined to the character or phone embeddings via addition or concatenation, often after the text encoder. Multi-lingual TTS models follow the same policy with language IDs [Y. Zhang et al., 2019].

Even though attitudes and emotions may be harder to identify in pre-recorded corpora, this challenge may be by-passed by explicitly recording acted corpora. The styles requested of the speaker are then used directly as labels, introducing Emotion/Style IDs as additional inputs

of the TTS [T.-H. Kim et al., 2020; Y. Lee et al., 2017]. Although this setup achieves state-of-the-art performances, the supervised training of a limited set of speaker or style embeddings raises two major issues:

1. This setup limits inference to the labels seen during training. Speaker/style embeddings are trained as a lexicon of categorical vectors which bias the decoder toward the style to apply. Without further regularization, this expressive embedding space is not ensured to be continuous, which prevents the inference with unseen pseudo-styles sampled from this space. Note that this issue may be partially mitigated for speaker embeddings by the introduction of pre-trained speaker embeddings from speaker recognition or conversion tasks[Chien et al., 2021; Jia et al., 2018][17].

2. The explicit recording of acted corpora requires controlled recording conditions, which is costly to set up, and may not scale to interactive natural dialogues, in which speaking styles are more nuanced and less caricatured.

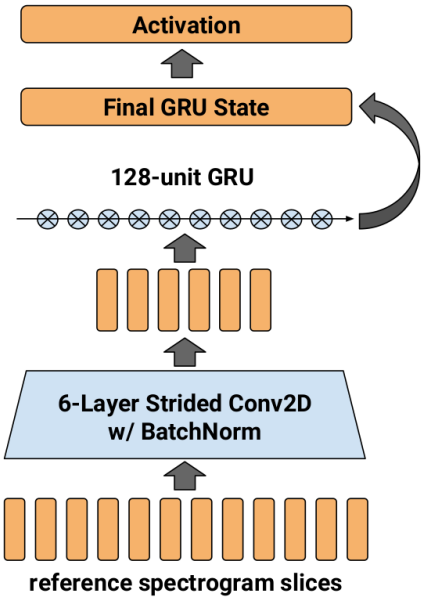
3. Speaker labels are more often than not correlated to channel information. The specificity of recording material used for each speaker is thus simultaneously learned by speaker embeddings. This entanglement of speaker and channel information results in varying voice quality among synthesized speakers.

1.2.2.2 Unsupervised Prosodic Representations

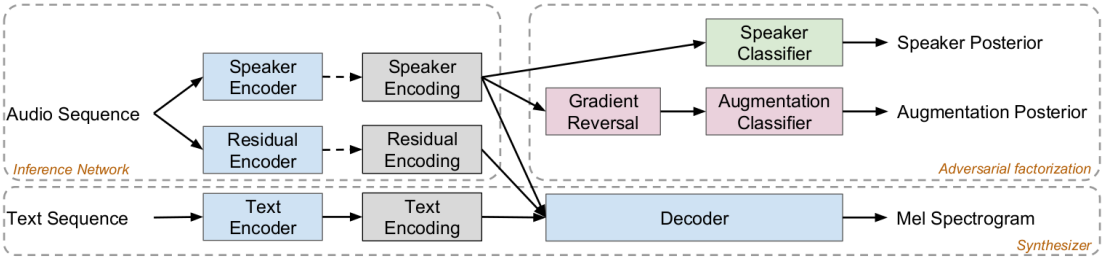
As a way to overcome the limitations of supervised expressive training of neural TTS, implicit approaches were proposed to encode the residual information left when text and speaker were already taken into account. On of the most decisive breakthrough in that regard is the *Reference Encoder* proposed by Skerry-Ryan et al. [2018]. This auxiliary neural module implicitly extracts a fixed-size vector which encodes the utterance-wise paralinguistic prosody of a reference audio signal. The summary vector is called *reference* embedding. The architecture of this reference encoder is illustrated in Fig. 1.8a. Skerry-Ryan et al. [2018] showed that this reference embedding could be used to achieve prosodic transfer toward the reference audio, in particular with non-matching textual contents.

This proposition of unsupervised training of prosodic embeddings from audio sequences enabled various breakthroughs in expressive modeling for TTS. Hsu et al. [2019] addressed the mentioned limitations of learnable speaker embeddings by introducing two classifiers on top of the reference encoder, following the architecture described in Fig. 1.8b. Through the use of adversarial-training, authors showed that the reference embeddings were able to learn various levels of information, including speaker and recording conditions. Nonetheless, the entanglement of paralingusitic features encoded into these reference embeddings needs to be addressed in order to envision these implicit representations as controllable biases in TTS.

[17] D-vectors [Variani et al., 2014] and x-vectors [Snyder et al., 2018] are used as speaker representations in this case.



(a) The Reference Encoder architecture.
Source: [Skerry-Ryan et al., 2018].



(b) Adaptation of the Speaker Embedding Layer with Adversarial Training. Speaker and residual encoders both follow the same architecture as the Reference Encoder. Source: [Hsu et al., 2019]

Figure 1.8: Integration of Audio References as TTS Inputs

1.2.2.3 Regularization of the implicit Prosodic Embeddings

The introduction of the reference encoder provides the opportunity to model paralinguistic prosody without the need for explicit expressive labels. However, the initial architecture proposed by Skerry-Ryan et al. [2018] requires an audio reference to constrain the output prosody. Such reference is not always available during inference, which is the reason why researchers have tried to enhance this neural layer in order to automatically structure these prosodic spaces through the extraction of meaningful directions. Y. Wang et al. [2018] proposed the Global Style Tokens (GST) architecture, illustrated in Fig. 1.9. The GST layer learns a limited set of utterance-wise style tokens (10 in the original implementation) which are supposed to learn disentangled expressive features from the reference embedding. During training, an attention layer computes the mixture of global tokens which composes the style embedding used to bias the output of the text encoder. During inference, the audio reference may be replaced by any

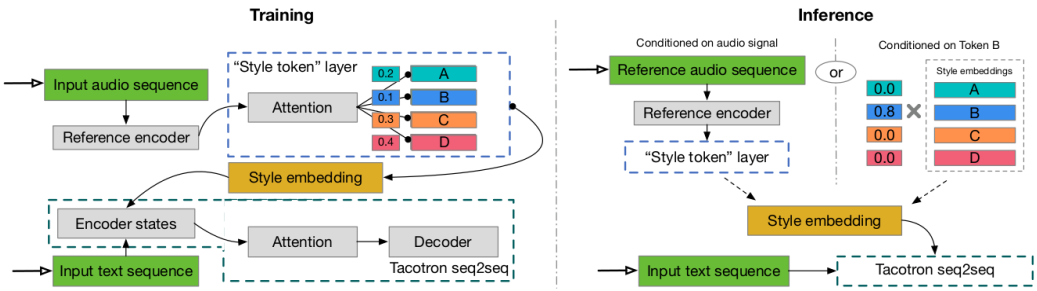


Figure 1.9: Integration of the Global Style Tokens layer in the TTS pipeline. Source: [Y. Wang et al., 2018]

hand-crafted mixtures of tokens. Similarly, Y.-J. Zhang et al. [2019] proposed a dimensional reduction of the reference embedding through a Variational Auto-Encoder (VAE). The authors showed that expressive features like mean-pitch, pitch variations and speaking rate could thus be controlled by sampling in the VAE latent space along individual dimensions.

Both propositions enable the expressive control during inference without the need for audio reference, assuming the post-hoc understanding of features encoded in the learned dimensions. Yet, the post-hoc exploration of such loosely constrained prosodic spaces may be quite costly: van Rijn et al. [2021] studied the controllability of the GST space [Y. Wang et al., 2018] through human sampling to reach a desired style. Although they found that participants managed to achieve the target style – which was recognized by independent listeners – this procedure is inherently model-dependent and requires several dozens of human participants. Therefore, this method cannot be applied in most real-life cases.

Sparse expressive labels have been shown to ease this exploration of prosodic embedding spaces. Sorin et al. [2020] visualized the reference embedding space with Principal Component Analysis (PCA) and found that distinct styles were projected in clusters in the first dimensions of the reduced space, even if these style labels are not exploited during the training. These findings advocate for endowing the reference encoder with the capability to learn prosodic representations disentangled from the textual content. These style vectors can then be used to control the expressiveness of synthesis at inference. Wu et al. [2019] showed that only 5% of labeled data were sufficient to train a Tacotron2-GST model with constrained tokens: the authors added a categorical cross-entropy loss to impose the global tokens mixture on a portion of the dataset. They found that each token thus learned one particular style that could be used during inference.

The post-hoc analysis of unsupervised paralinguistic embeddings offers a promising means of combining the best of both worlds, since the identification of meaningful directions in the unconstrained (or loosely constrained) embeddings finally provides an interpretation of the features encoded into these neural representations. The understanding of these factors is crucial to turn these unsupervised latent spaces into effective control interfaces, while taking advantage of the extrapolation capabilities of continuous unsupervised representations.

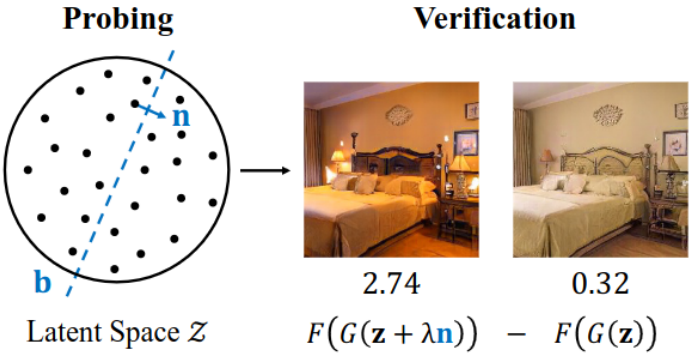
1.2.3 Exploration of Neural TTS Embeddings

Despite the relative success of the presented methods of bias introduction to capture and control the numerous degrees-of-freedom of expressiveness in speech, little is known about the specific features learned by these internal representations. The difficulty of disentangling attitudes in natural speech recordings is a burden to the objective analysis of the correlation between acoustic cues and expressive styles. The training of specific attitude embeddings – either in a supervised or unsupervised manner – provides a way to analyze style productions individually from a statistical perspective. The better understanding of neural embeddings thus represents a step forward toward the use of speech technologies as a statistical analytic tool for speech sciences, as will me demonstrated in Chapter 4. Moreover, the understanding of intermediate computations performed by neural layers is required for designing more careful control methods for neural architectures, as illustrated in Chapter 5.

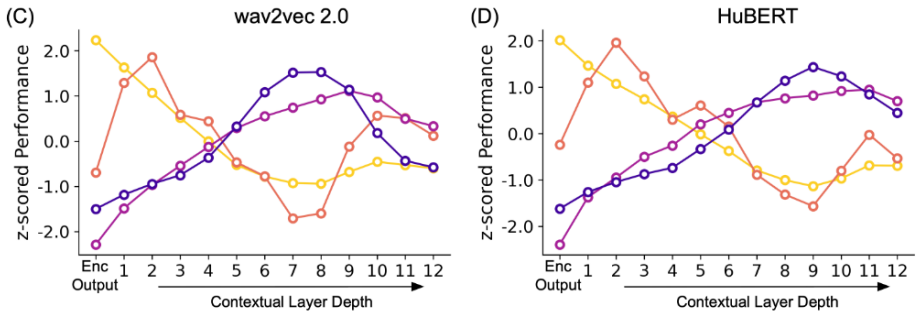
1.2.3.1 Probing Embeddings in Other Fields

Although the probing of features encoded in internal embeddings of neural models remains quite rare for TTS models, enlightening work from other fields has shown the benefits of a closer analysis of these intermediate representations. Zeiler and Fergus [2014] have shown that successive CNN layers trained on image classification learned to recognize more and more complex patterns in input images. The finer understanding of the task performed by each layer helped the authors to design a more thoughtful architecture of their model, resulting in better performance in their recognition task. In the field of Neural Machine Translation, Bau et al. [2019] identified the neurons which encode gender and tense. They showed that a bias on these neurons could provide control of these categorical features when translating between languages in which gender or tense are implicit. Similarly, Yang et al. [2021] exhibited directions in Generative Adversarial Network (GAN) latent space for image generation that encode the lighting of scenes, types of room or layouts (see Fig. 1.10a). They moreover show that the control of the room lighting through biases along the found direction requires lights to be switched-on: like pauses and speech rate, categorical and continuous control of desired dimensions are naturally bounded.

In the field of audio signal analysis, Vaidya et al. [2022] and Wells et al. [2022] have shown how phonetic and acoustic features were represented in successive layers of self-supervised neural models like wav2vec [Baevski et al., 2020; Schneider et al., 2019] and HuBERT [Hsu et al., 2021] (see Fig. 1.10b). They found that middle and end layers best encode high-level phone and word features respectively, while first layers encode low-level spectral features. These findings indicate which intermediate representations to use in sub-tasks depending on the type of information needed.



(a) Causal Control of lighting in closed room with GAN network. Source: [Yang et al., 2021].



(b) Probing by layer of speech units in Self-Supervised Audio Embeddings. Source: [Vaidya et al., 2022].

Figure 1.10: Examples of Exploration of Intermediate Embeddings in the Literature.

1.2.3.2 Analyzing Embeddings in TTS

In comparison with the abundant literature on neural TTS, relatively few studies actually took a closer look at the features encoded into intermediate embeddings. Table 1.4 gives a list of such studies I am aware of, with a focus on work exhibiting specific features or exploring latent spaces with innovative methods. Three types of representation emerge:

- **Phone and Character embeddings:** As atomic building units of TTS systems, phone and character embeddings play a central role in the presented architectures. However, little is known on how these representations are structured, how they relate to each other and what other features or units they encode if any. Internal representations of orthographic inputs sequences were found to encode corresponding phonemes in a French Tacotron [Perquin et al., 2020] (see Fig. 1.11a). Although that study only provides qualitative observations of the phonetic space, it constitutes a first step into better understanding TTS embeddings.

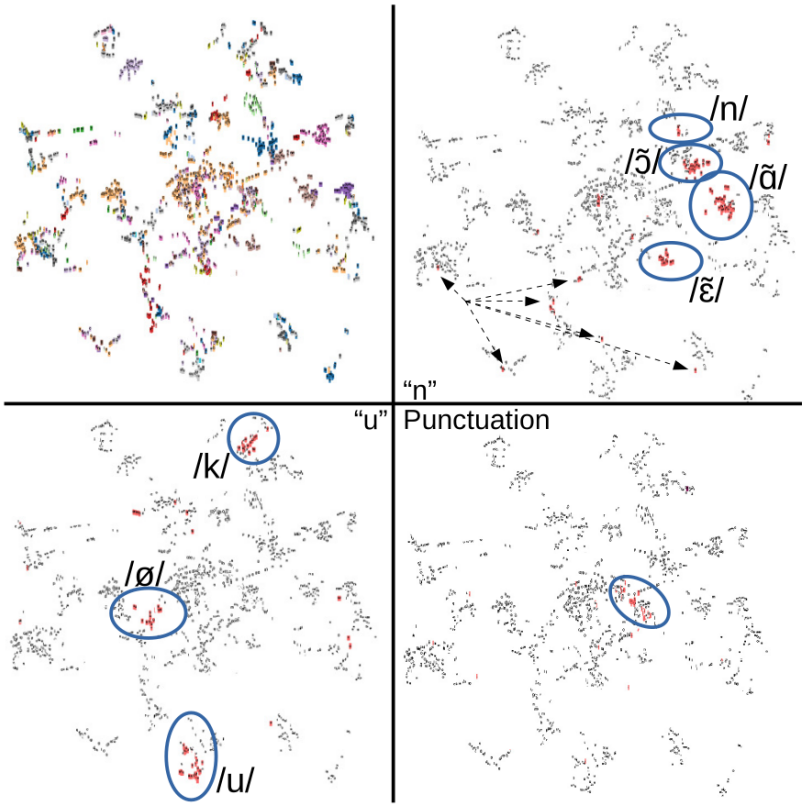
Reference	Baseline TTS	Granularity	Comb	Features Found	Sampling in latent space
Perquin et al. [2020]	Tacotron	Character	✗	Phonetic	✗
Jia et al. [2018]	DeepVoice	Speaker	⊕	Gender	Hand crafted pseudo-speakers
Skerry-Ryan et al. [2018]	Tacotron	Speaker / Style	⊕	✗	✗
Hsu et al. [2019]	Tacotron2	Speaker / Style	⊕	Gender / Noise conditions	✗
Shin et al. [2022]	Tacotron2	Speaker / Style	+	✗	Unseen Style Tag applied
Y. Wang et al. [2018]	Tacotron	Style	+	Pitch / energy / speed	Hand crafted GST mixtures
Wu et al. [2019]	Tacotron2	Style	+	✗	✗
Y.-J. Zhang et al. [2019]	Tacotron2	Style	+	Pitch / speed	Interpolation in VAE space
Sorin et al. [2020]	Tacotron2	Style	⊕	✗	Centers of reference emb
M. Kim et al. [2021]	WaveGAN	Style	?	✗	Unseen Style Tag applied
Tits et al. [2021]	DC-TTS	Style	⊕	eGeMAPS	Human sampling
S. Wang et al. [2017]	✗	Speaker	✗	Identity / Gender	✗

Table 1.4: Various granularity of Embeddings Analysis in the TTS Literature. "Comb": Combination, ⊕: Concatenation,+: Addition.

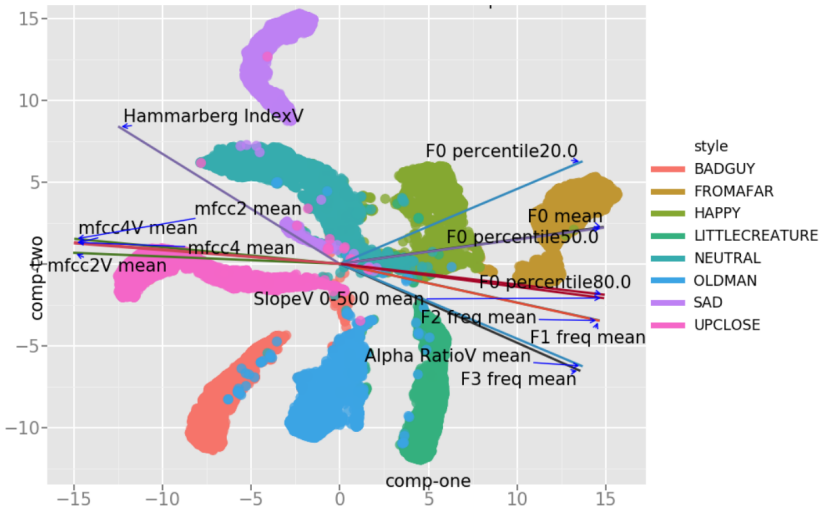
- **Speaker embeddings:** S. Wang et al. [2017] have shown that speaker embeddings learned for speaker verification encode speaker identity and gender, but also speech content and channel information, which are likely to interfere with the linguistic content encoded by the text encoder. These pre-trained speaker embeddings from verification tasks have been successfully used in TTS as a way to mitigate the limitation to seen speakers during inference [Jia et al., 2018].

- **Style embeddings:** As a method to explore acoustic features in utterance-wise style representations, Tits et al. [2019] and Tits et al. [2021] proposed to identify mean acoustic parameters by utterance in reference embeddings. Their method consists of exhibiting directions in the latent space corresponding to continuous acoustic parameters variations, as illustrated in Fig. 1.11b. They found high correlations between acoustic parameters measured on the audio and predicted from the latent space (e.g., R=0.72 for fundamental frequency median). Although this analysis is motivated by the desire to create an interpretable style control interface, the authors have not used this identification of acoustic parameters in their proposed interface.

Similar methods were used by Cho et al. [2023] to track articulatory trajectories in self-supervised audio representations. They trained linear predictors from embeddings extracted from various state-of-the-art self-supervised learning (SSL) models (wav2vec [Baevski et al., 2020], HuBERT [Hsu et al., 2021], etc.) and computed how accurate the feature predictions were from each model. This linear probing method is an interesting approach to investigate the variety of features potentially encoded into neural embeddings, that we also employed in our work in Chapter 4.



(a) t-SNE visualization of character embeddings contextualized by the Tacotron encoder. Source: [Perquin et al., 2020].



(b) Expressive Latent Space annotated with directions of variation of acoustic features. Source: [Tits et al., 2019].

Figure 1.11: Visualization of TTS embeddings at character-level (left) and expressive utterance-level (right).

1.3 Evaluation

Neural TTS models have reached such high standards that the speech they generate is difficult to distinguish from natural voice. In 2018, Shen et al. [2018] reported that their Tacotron2 model had already reached Mean Opinions Scores (MOS) of 4.53/5 for speech naturalness, compared with 4.58/5 for a voice recorded under professional conditions. In latest Blizzard Challenge 2023, two models achieved MOS comparable with the natural voice on the quality assessment of the Hub Task [Perrotin et al., 2023][18]. Even more surprising, four models were judged statistically closer to the reference speaker than the natural speech in the speaker similarity evaluation of the Spoke Task.

These performances question the evaluation of synthetic models: are common experimental setups (summarized in Table 1.5) suitable for assessing such subtle nuances between models? How does the wording of instructions given during perceptual tests affect listeners judgment? What role does objective evaluation play in the assessment of synthetic voice quality? This section discuss the evaluation of synthetic models and their comparison to natural voices. Objective and perceptive evaluations are discussed in subsection 1.3.1 and subsection 1.3.2 respectively. The limitations of both types of evaluation question the evolution of the evaluation framework: we discuss our thoughts on this challenge in subsection 1.3.3.

Table 1.5: Most common evaluation metrics of synthetic voices in the literature.

Evaluation	Metric	Abbreviation	Scale	Goal
Objective	Word Error Rate	WER	[0; 100]	Intelligibility
	Mel Cepstral Distortion	MCD	[0; +inf]	Averaged Comparison with Ref
	Mel Spectral Distortion	MSD	[0; +inf]	Averaged Comparison with Ref
	Mean Squared Error	MSE	[0; +inf]	Comparison with Ref on specific feature
	Mean Absolute Error	MAE	[0; +inf]	Comparison with Ref on specific feature
	F0 Frame Error rate	FFE	[0; 100]	Pitch errors on voiced frames
Subjective	Mean Opinion Score	MOS	[1; 5]	Absolute Likert Scale
	MUSHRA	MUSHRA	[0; 100]	Comparative Likert Scale
	Comparative MOS	CMOS	[−3; +3]	Preference Scores
	AB(X) Preference	AB(X)	Binary ratio	Preference

1.3.1 How to objectively evaluate synthetic speech?

In the supervised deep-learning protocol, objective metrics are necessarily used in order to compute the training losses of neural models. All TTS models are trained in order to minimize (at least) the error between the target and the predicted output spectrograms (resp. waveforms in case of fully end-to-end text-to-wav models). However, the optimization of this spectral loss does not always correlate with better perceptual judgments. This may be explained by a few factors: 1) the range of variations of objective measurements may not be perceptible by non-expert listeners. 2) most models are trained under the teacher-forcing method. This

[18]This result is mitigated by Perrotin et al. [2023]: a MUSHRA was performed with the best models, which exhibited significant differences between the natural voice and the synthetic models.

means that during training, model's predictions are partially replaced by ground-truth values when re-injected in the model[19]. As a result, training losses are not evaluated in the same context as will be experienced at inference time. This mismatch between losses and perceptual evaluation is an issue for reproducible training of TTS models. The minimization of training losses does not guarantee that the model has reached its best perceptual quality, which also prevents the use of early-stopping [Prechelt, 2002]. Training losses may not provide significant insights into the perceptual quality of TTS models, and are thus mostly not reported in the literature.

The objective evaluation of the synthesized test set is not trivial either. Most of the metrics reported in the literature compute a difference between two stimuli, generally between the model's prediction and the corresponding ground truth. This setup evaluates synthetic voices on the task they have been trained on. The recorded voice is considered as the best achievable quality which the synthetic voice should mimic. However, as described in Section 1.2.1, speech can take many forms depending on the context, the speakers or their communicative intent. The evaluation of isolated sequences in comparison with an arbitrary reference cannot fully assess the quality of the synthesized speech. Nonetheless, objective measurements can complement perceptual ratings as a quantitative evaluation of the underlying factors of subjective judgments.

1.3.2 Perceptual Assessment of Synthetic Voice

The most commonly-used perceptual metric is the Mean Opinion Score (MOS) of speech naturalness. MOS tests consist in asking participants to rate individual stimuli on a half-point or full-point scale between 1 (very poor) and 5 (excellent) on a Likert Scale [van Heuven & van Bezooijen, 1995], and averaging these ratings over several stimuli to evaluate each synthesis system. Evaluating the naturalness of a voice involves paying attention to several aspects: are the phonetics correctly pronounced? Is the voice expressive or monotone? Is the spectrum similar to that of a human voice? Therefore, the underlying factors of people's ratings of voice naturalness are numerous: clarity and intelligibility play an important role, but also accent and tone [Shirali-Shahreza & Penn, 2023]. Clark et al. [2019] demonstrated that the duration of the stimuli also impacts perceptual judgments: longer speech forms likely provide additional linguistic context which helps participants to build more precise expectations [Latorre et al., 2014].

The ambiguity of participants' ratings is further reinforced by the lack of precision of the instructions given by researchers in their evaluation setup. Kirkland et al. [2023] reported that most studies published in Interspeech and SSW papers in the 2021-2022 period do not describe their evaluation setup in a reproducible manner. While this does not strictly indi-

[19] As an example, in Tacotron2 training [Shen et al., 2018], the end of sequence is triggered when the generated spectrogram reaches the length of the target, instead of using the end of sequence predictor. Additionally, ground-truth mel frames are passed through the prenet instead of predicted frames. Similarly for FastSpeech2 [Ren et al., 2021], predicted prosodic features are replaced by ground-truth values during the training.

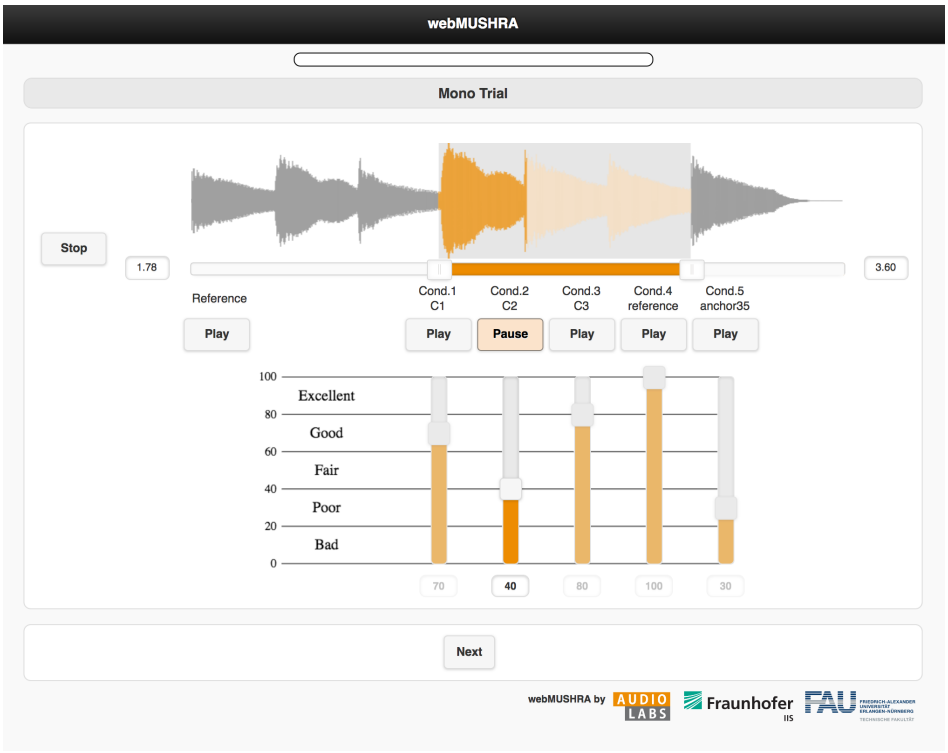


Figure 1.12: The webMUSHRA interface: several stimuli are displayed simultaneously. The reference is explicitly given, and also hidden among the stimuli. The hidden reference provides an example of optimal stimulus. The condition labels are hidden during the experiment. Note that the waveform displayed in this example is not a speech recording and only stands for illustration purposes. Source: [Schoeffler et al., 2018]

cate that experiments were not designed carefully, it surely hints that not enough attention is given to the use of MOS in TTS assessment. Moreover, experimental factors like the rating's scale [Kirkland et al., 2023] and wording [O'Mahony et al., 2021] have been found to significantly impact participants ratings of naturalness. These findings question the relevance of naive MOS as a perceptual metric for synthetic speech evaluation.

Comparative evaluation setups like Comparative MOS (CMOS) and MUSHRA [International Telecommunications Union, 2003] present listeners with a slightly more precise task. Synthetic voices are here evaluated in comparison with each others or against a natural reference, which at least ensures that participants provide a hierarchical rating of the proposed systems, regardless of their personal rating factors. The MUSHRA test is illustrated in Fig. 1.12. The original MUSHRA setup may even be adapted in this direction to reinforce distinctive ratings between systems [Kayyar et al., 2023]. A high anchor reference is both hidden (to rate) and explicitly provided in the MUSHRA setup. It provides participants with an undisputed mental image of the goal to achieve [Latorre et al., 2014]. However, as stated for objective measurements, such undisputed reference does not always makes sense in the evaluation of speech. With the increasing focus on speech prosody and expressiveness, even the natural voice (when available) is not the only valid example of speech production in the evaluated

context. Due to the variability of expressive nuances in speech production, there is no single golden standard for narration or the production of speech attitudes [Bachorowski, 1999]. As a result, the evaluation of prosodic transfer toward a reference sample may not really assess the quality of expressive TTS.

1.3.3 Evolution of the TTS evaluation framework

Despite the known limitations of the MOS evaluation, this is still the most common way to evaluate synthetic voices. Gutierrez et al. [2021] described several reasons for this default choice: MOS tests are easy to set up and require less cognitive load than multi-ranking tasks like MUSHRA. MOS of speech naturalness is so widely used in the literature that it has become an implicit requirement to compare new models to existing baselines. MOS is not inherently useless evaluation, but it is certainly not exhaustive. MOS should be designed with a particular attention to the wording, and this design should be reported to enable reproducible studies.

One simple extension of the single MOS test would be to expand this rating to different scales associated with various more explicit and precise instructions. As an example, Hinterleitner et al. [2011] ran a two-stage pretest to isolate a set of 16 attribute pairs that were relevant in the description of synthetic signals. These factors include: artificial vs. natural, unnatural rhythm vs. natural rhythm, unpleasant vs. pleasant, distorted vs. undistorted, etc. Authors asked participants to evaluate each attribute independently. Their analysis revealed that three perceptual dimensions mainly explain participant appreciation of synthetic speech: naturalness, spectral disturbances and temporal distortions. Asking participants to rate these three aspects on three separated MOS scales would provide more insights on listeners perception of new proposed models. This multi-scales evaluation framework was applied by King and Karaiskos [2013] in the Blizzard Challenge 2013. These underlying dimensions may be inferred afterward by multidimensional projection methods like Multidimensional Scaling (MDS) [Kruskal & Wish, 1978], but the meaning of these unveiled dimensions is subject to interpretation [Mayo et al., 2005].

Objective evaluations should be systematic: listeners' preference is the main goal, but evaluations should provide cues to understand the reasons for this preference in the model generative process. The issue with objective metrics is that they often provide a general observation of the measured phenomena, which tends to hide the interesting points of divergence between evaluated models. Now that state-of-the-art models have reached high standards, neural TTS systems produce very similar speech most of the time. As a result, perceptual judgments may be mostly correlated to outliers [Osborne & Overbay, 2004]. Thus, utterances must be selected carefully for listening tests: a random selection of utterances is likely to include a large portion of samples from the core distribution of models capabilities. To mitigate this effect, test utterances may be selected based on their distinctiveness according to an objective measurement, such as spectrum MSE for example, as performed in the Blizzard Challenge 2023 [Perrotin et al., 2023].

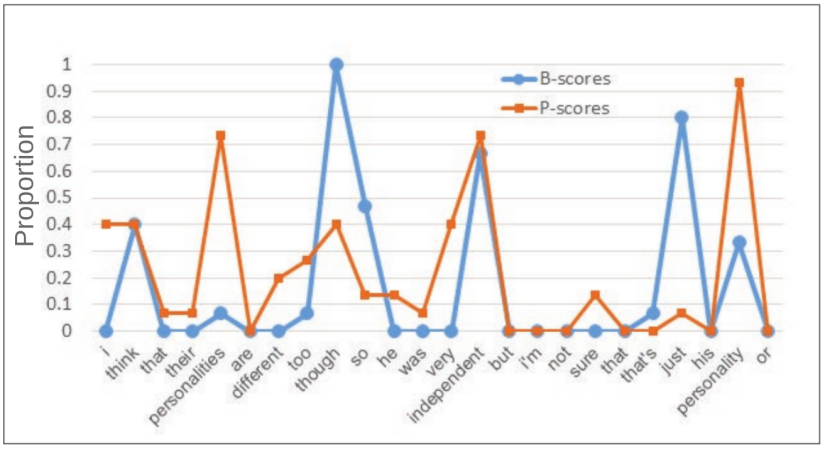


Figure 1.13: Illustration of the RPT at word-level: P-scores (resp. B-scores) indicate the proportion of participants who annotated a word as prominent (resp. as preceding a boundary). This framework can be adapted to target the portions of the utterance which were decisive in listeners' judgments. Source: [Cole & Shattuck-Hufnagel, 2016]

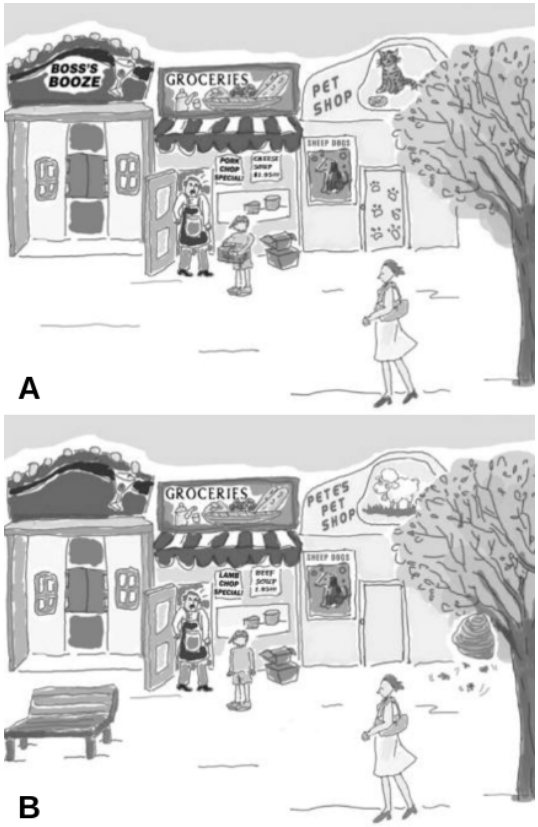
Alternatively, in order to target smaller portions of interest in evaluated stimuli, Gutierrez et al. [2021] proposed an evaluation interface which integrates the Rapid Prosody Transcription (RPT) method [Cole & Shattuck-Hufnagel, 2016]. This method enables explicit targeting of speech units (mostly words, but the method could be adapted to portions of audio signals) which listeners have judged of particular interest with regard to the given instruction. This method provides a way to automatically isolate smaller portions of audio signals based on crowd-sourced data, in order to exhibit objective differences between models on the specific portions which were decisive in participants' judgments. Fig. 1.13 illustrates RPT at word-level.

The design of more challenging experiments would also provide more varied examples of the specific contexts in which synthetic models diverge. In the case of intelligibility for example, the organizers of the Blizzard Challenge 2023 specifically designed a heterophonic homographs disambiguation task from orthographic inputs [Perrotin et al., 2023]. There are 789 homographs in French[20], but most pairs are not evenly balanced in natural corpora. As a result, synthetic models may be biased to only produce the most seen variant. However this bias would not produce any error in classical evaluation setups since the other variant would not appear in the test set. The specific design of challenging evaluation sets for a wider variety of tasks thus provide better comparisons between systems.

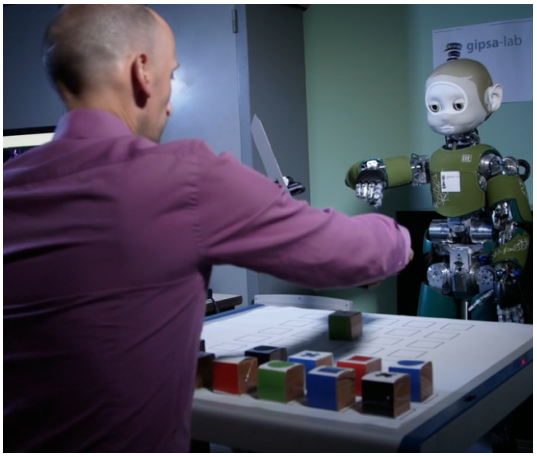
Finally, synthetic voices are now integrated in interactive systems. In order to design synthetic voices that would benefit to their use-case, another way to evaluate TTS would be to evaluate their performance during an interactive task with a human participant [Wagner et al., 2019]. The Diapix task [Van Engen et al., 2010] or true physical task-oriented conversations [Skantze, 2021] are examples of such interactive tasks which require the collaboration between humans and artificial agents through efficient communication. These tasks are illus-

[20]Source: https://fr.wiktionary.org/wiki/Catégorie:Homographes_non_homophones_en_français

trated in Fig. 1.14. Synthetic voices may substitute for one of the participants'voices using a Wizard-of-Oz setup [Dahlbäck et al., 1993] in order to evaluate how synthetic speech features impact collaborative performance.



(a) Example of Diapix pair. Each participant can only see one of the two pictures. Participants have to collaborate to find all the differences between the pictures A and B. Source: [Van Engen et al., 2010].



(b) Multi-Party Interaction between Nina and a human participant. Source: https://images.cnrs.fr/video/6585

Figure 1.14: Examples of Collaborative Tasks.

1.4 Our Contributions

This chapter presented the main challenges of the prediction of speech from text, and the methods implemented by state-of-the-art neural models to produce synthetic speech which is comparable with natural speech recordings. This chapter also reviewed the specific challenges of modeling and evaluating expressive cues in speech.

Our work aims at bridging the gap between generative speech technology and speech science in an attempt to understand the modeling of phonetic and phonological features in the internal representations of neural TTS models. By providing analytic methods to probe

the intermediate embeddings of neural TTS, our goal is to provide data-driven interpretations of the underlying processes performed by these deep-learning algorithms. We believe that the semi-supervised statistical analysis of natural speech data performed by TTS models in order to maximize their predictive capabilities could provide new insights on the underlying mechanisms of speech production. Moreover, the breakdown of operations performed by each layer of these deep-algorithms enables us to design more careful expressive control mechanisms.

More specifically, our contributions revolve around the modeling of two layers of prosody:

Linguistic prosody: We study **linguistic prosody** in Chapter 3 through the exploration of alternative ways to present the training corpus to the neural model. The segmentation and the augmentation of textual inputs with initial punctuation marks are evaluated to that purpose. The linear probing of acoustic features in TTS intermediate embeddings in Chapter 4 resulted in our proposition of explicit control of acoustic features through linear biases in Chapter 5. This control is a first attempt to infer **propositional attitudes** through acoustic control. To the best of our knowledge, this method is the first attempt to control the generative process of TTS models from post-hoc internal space analysis, called **Causal Control**.

Paralinguistic prosody: This control of **propositional attitudes** is further enhanced through the introduction of an auxiliary neural layer called Local Style Tokens in chapter 6. These attitudes, indistinctly referred to as **styles**, are seen as a way to enhance the synthetic voice message with contextual information conveyed by the paralinguistic prosody. Through the introduction of local prosodic contributions, we emphasize the specific interaction between propositional attitudes and the textual content itself [Beckman & Pierrehumbert, 1986; Selkirk, 1986].

Two models were selected to implement these approaches: **Tacotron2** [Shen et al., 2018] and **FastSpeech2** [Ren et al., 2021]. Despite the promising performances of stochastic models mentioned in Section 1.1.5, Tacotron2 and FastSpeech2 were chosen for their prevalence in the expressive TTS literature. That being said, the contributions described in the following chapters could similarly be applied to other architectures. The training of our initial baseline models on which we implemented the following contributions is further detailed in Chapter 2. This chapter also addresses the question of the most suitable input representations. As mentioned in Section 1.1.1, using phonetic transcriptions as TTS input restrains the ability of end-to-end models to capture and explain variability of text renderings by speech. However, the training on orthographic inputs alone can be challenging for neural models. We instead explored the use of mixed representations to combine the best of both worlds [Kastner et al., 2019].

Throughout the following contributions, we pay a particular attention to the evaluation of the proposed systems. Comparative setups like MUSHRA and CMOS are preferred for their more specific task which generates less confusion among listeners. These perceptual ratings are further explored through post-hoc multidimensional analysis when relevant. These evaluations are always accompanied by objective metrics in order to better interpret the underlying factors of perceptual judgements.

Chapter 2

Design of our Baseline French TTS

Contents

Chapter Highlights

This chapter presents our baseline French TTS systems based on two state-of-the-art models: Tacotron2 and FastSpeech2. Both models were modified to mitigate their main limitation. 1) We implemented a correction mechanism for Tacotron2, called **Gate Loss Correction**, which ensures the correct detection of End-of-Sequences. 2) We proposed a **new Letter-to-Sound (L2S) alignment** inferred from the observation of attention maps from Tacotron2. This L2S allowed us to **train FastSpeech2 on orthographic inputs**, without the need for an external phonetizer front-end. This L2S was used to align the corpus which was used in all following contributions. We entered the **Blizzard Challenge 2023** with our orthographic-enhanced FastSpeech2 baseline.

Related contributions: [Bailly et al., 2023; Hajj et al., 2022; Lenglet et al., 2022a, 2023c]

As described in Chapter 1, state-of-the-art neural TTS models are able to generate synthetic speech that is very close to natural recordings. Two models have been given a particular focus for their prominence in the expressive TTS literature: Tacotron2 [Shen et al., 2018] and FastSpeech2 [Ren et al., 2021]. Although sharing a similar encoder/decoder architecture, FastSpeech2 replaces the convolution and recurrent LSTM layers [Hochreiter & Schmidhuber, 1997] used in Tacotron2 by multi-head self-attention layers [Vaswani et al., 2017]. Self-attention layers are favored in FastSpeech2 for their capacity to model contextual information without suffering from LSTM difficulties in modelling long-term dependencies [Hochreiter et al., 2001].

In addition, FastSpeech2 replaces the attention network [Chorowski et al., 2015] used by Tacotron2 to learn the alignment between the input sequence and the output mel-spectrogram by a duration predictor. This duration predictor is trained to predict the number of spectrogram frames corresponding to each phone of the input sequence. This allows FastSpeech2 to predict the length of the output sequence before starting the decoding process. In doing so, FastSpeech2's decoder generates all spectrogram frames in parallel. In contrast, Tacotron2's decoder is autoregressive, which leads to slower inference and training. Moreover, FastSpeech2 implements two prosodic predictors for pitch and energy. These additional tasks have been shown to improve the perceived synthetic voice quality, and provide explicit control of these prosodic features at inference.

As stated by Shen et al. [2020], while being slower, autoregressive decoders contribute to the naturalness of the audio output. As a result, both architectures present specific advantages and drawbacks depending on the use-case. Tacotron2 is inherently easier to train, since no preprocessing of the training data is required in order to compute time-alignments of the input sequences nor to extract prosodic features. In return, the unsupervised attention mechanism is not as robust as the duration predictor of FastSpeech2. To mitigate this issue, we proposed to modify the training process of Tacotron2 to fine-tune the model with a specific focus on the prediction of End-of-Sequence, called **Gate Loss Correction (GLC)**.

Although the duration predictor of FastSpeech2 avoids the mentioned artifacts, this predictor is also a limiting factor when training FastSpeech2 on orthographic input. Indeed, the time-segmentation of the training set, necessary to train this predictor, is unclear when processing orthographic sequences. As a result, TTS models which implement an explicit duration predictor are generally trained solely on phonetic inputs. In this thesis, we advocate instead for the use of orthographic inputs as elementary building blocks of speech units (see Section 1.1.1 for further details). That is why we proposed a new Letter-to-Sound (L2S) alignment inferred from the observation of attention maps of a pre-trained Tacotron2 model. We used this alignment to assign duration to the orthographic sequences in order to train a FastSpeech2 model on orthographic inputs. Moreover, we showed that the addition of a phonetic prediction task from the output of the FastSpeech2 text encoder allows us to train the model on <orthography|phonetic> pairs without the need for audio recordings. This setup helps learning phonetic transcriptions for words and contexts that are otherwise rarely found in classical audiobooks training corpora.

This chapter describes the proposed implementation modifications to Tacotron2 and FastSpeech2, used as baseline models for our later contributions. The implementation of Tacotron2 is detailed in Section 2.1. The successful training of Tacotron2 allowed us to design our L2S alignment from the analysis of attention maps as described in Section 2.2. This alignment was used to demonstrate the benefits of training FastSpeech2 on both orthographic and phonetic representations. We entered the Blizzard Challenge 2023 with our proposed mixed-input FastSpeech2: our model and its results are detailed in Section 2.3.

2.1 Implementation of our Tacotron2 Baseline

The relative ease of training of Tacotron2 made us consider this model for our first experiments. This section presents our preliminary work to mitigate the attention flaws reported on the original Tacotron2 attention mechanism. We then show how the observation of the attention maps computed by this Tacotron2 highlights the statistical regularities learned by the text encoder to produce phonetic sequences from orthographic entries. We inferred L2S rules from these observations, which allowed us to enhance the implementation of FastSpeech2 with orthographic inputs, as described in Section 2.3.

2.1.1 The Challenging Training of Attention on Orthographic Sequences

The main issue noticed during preliminary work with Tacotron2 is the unreliability of the attention mechanism. When no additional constraints are given to the attention layer, artifacts may appear during inference. The most prevalent artifacts are omissions or repetitions of words or syllables, and the production of unintelligible speech due to the mixing of phones within one word. Repetitions are especially common in short utterances (less than 2 s), while omissions are more frequent in utterances longer than 10 s.

Solutions have been proposed in the literature to avoid these artifacts. Non-attentive Tacotron2 [Shen et al., 2020] entirely replaces the attention mechanism and the End-Of-Sequence (EoS) by a duration predictor. Similarly to FastSpeech2, this method requires the phone durations of the corpus for the training. Other propositions implement additional constraints on the attention mechanism, such as the Monotonic Attention [He et al., 2019]. This method is better suited for phonetic inputs, in which all the elements of the input sequence do have to be pronounced from left to right. On the other hand, the monotonicity of the alignment path is not ensured for orthographic sequences: opaque languages like French or English lack a direct mapping between characters and the audio sequence to produce. Fluent reading of French requires to process several characters in one fixation [Bosse & Valdois, 2009].

Fig. 2.1 reports the range of characters to take into account in order to utter a character correctly in French. Bosse and Valdois [2009] computed the visual attention span (VAS) required on a set of entries from the Robert dictionary, augmented with part-of-speech (POS) tags. This analysis of a decision tree to compute letter-to-sound (L2S) mapping from a set

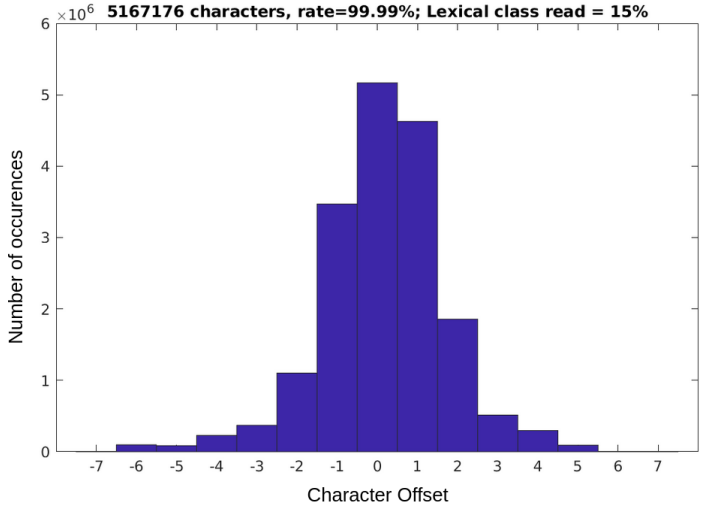


Figure 2.1: Visual Attention Span needed to process French orthographic sequences. Source: [Bosse & Valdois, 2009].

of 500 000 French words shows that up to 5-6 characters ahead and behind are needed to compute the correct phonetic mapping. Note that the decision tree questions the POS tag of the word 15% of the time.

In order to similarly allow the attention mechanism to look ahead in the sequence to determine the appropriate phone to produce, and to eventually skip characters that should be muted, the monotonic constraint was not introduced in our model.

2.1.2 Fine-Tuning of the End-of-Sequence Detection of Tacotron2

Instead of imposing the monotonic constraint of the attention layer, we hypothesize that the attention mechanism artifacts were generated because of the mispredictions of the End-of-Sequence (EoS) predictor. Early stopping of the autoregressive process tends to elide syllables, which are not necessarily the last syllable of the utterance, because end of sequences are anticipated by the model through the introduction of later context by the bi-LSTM in the text encoder. Similarly, in absence of detected EoS for shorter utterances, the attention layer focuses on input characters which may have already been pronounced, resulting in skipping, repeating or attention collapse (unintelligible gibberish). This hypothesis is supported by the frequent lack of detection of EoS reported by He et al. [2019].

In order to reduce the mispredictions of EoS, we added a systematic fine-tuning process to Tacotron2. During this fine-tuning, two modifications are introduced, illustrated in Fig. 2.2. First, 9 frames of recorded room tone are added at the end of each utterance, during which the EoS probability is set to 1. This silence comes from the pause following each utterance in the original recording. Second, a multiplying factor λ is added to the gate loss error before back-propagation, following equation. 2.1. The rest of the model is not frozen during this fine-tuning step. This process is called **Gate Loss Correction (GLC)**. We empirically found that

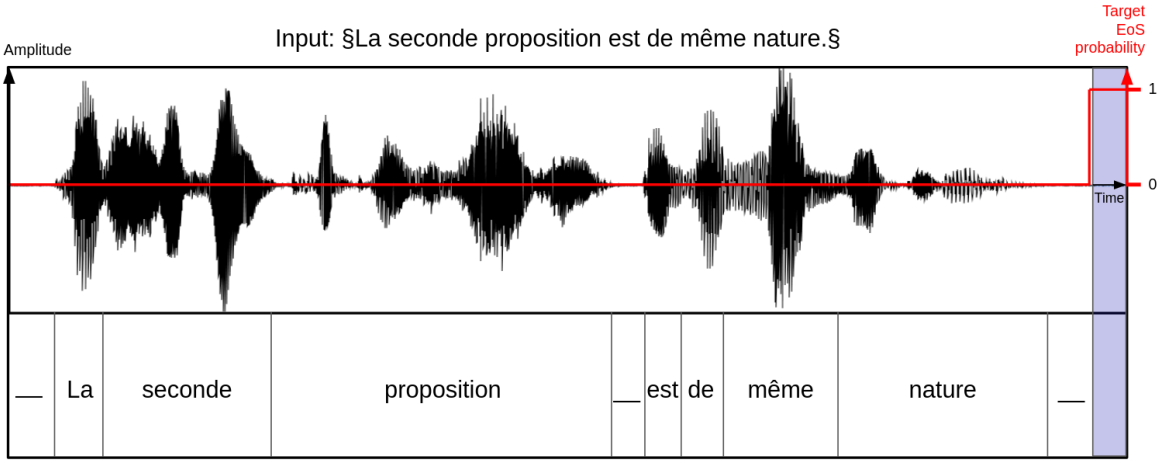


Figure 2.2: Illustration of the proposed fine-tuning process of the End-of-Sequence (EoS) Prediction for Tacotron2. During the fine-tuning, 9 frames of silences (in blue) are added at the end of the utterance.

these modifications correct previously mentioned artifacts, and improve the overall synthesis quality. Various setups were tested with the GLC: best results were found for the training procedure described in Appendix B.1. The benefits of these modifications are evaluated in Section 3.2.

$$L_{TC} = L_S + L_{PS} + \lambda * L_G \tag{2.1}$$

with L_{TC} the total loss of Tacotron2, L_S the MSE spectral loss, L_{PS} the MSE spectral loss after the Postnet, and L_G the cross-entropy Gate Loss. λ is initially set to 1, and is increased during the fine-tuning of the EoS prediction.

2.1.3 Changes in Model Configuration

Additionally, two changes have been made to the original implementation of Tacotron2:

1. Following Zen et al. [2016], the autoregressive decoder is trained to predict two frames per step instead of one. This process speeds up training and inference. This is equivalent to the prediction of 160 mel-coefficients instead of 80 from the linear projection of the autoregressive decoder. The prenet is therefore adapted to account for this additional input at each decoding step. This change does not impact the postnet, which is only applied after the prediction of the entire mel-spectrogram. Our preliminary studies found no noticeable degradation in audio quality when generating two frames at once instead of one.

2. The dimension of the prenet is reduced from 256 to 128. As a reminder, during the training phase, the mel-spectrogram frames predicted by the decoder are replaced by Ground-Truth data[1]. As stated in Section 1.1.1, mel-spectrograms were chosen as output speech representations for their smoother time-domain variations. Implicitly, this means that copying the mel-coefficients from the last computed frames would be a valid strategy to minimize the spectral loss of the model. The prenet acts as a bottleneck to prevent this strategy. We found that the reduction of the prenet dimension from 256 to 128 was necessary to achieve this goal.

We also adapted the mixed-representations framework proposed by Kastner et al. [2019]. In the original framework, characters and phones can be used simultaneously to train a neural TTS model. Such combination of input types have been showed to benefit both types of representations during inference. In the original framework, each word is given a 50% chance to be either transcribed with phonetic or orthographic symbols. We adopted a slightly different approach: instead of mixing representations in each utterance, the corpus is presented to the model twice by epoch, once using orthographic inputs and once using phonetic inputs. Thus, we ensure that the model is trained on a maximum number of different words transcribed with orthographic symbols, which maximizes the model performances on orthographic inputs at inference.

Other hyperparameters are kept unchanged from the original Tacotron2 implementation. The exhaustive list of hyperparameters and the default training procedure we used in the presented experiments are summarized in Appendix B.

2.2 Letter-to-Sound Alignment from the Attention Mechanism

As detailed in Section 1.1.3.1, the text-audio alignment is a core feature of sequence-to-sequence TTS models. Usually, phone durations vary between 50 ms and 150 ms. Following the most common computation of mel-spectrograms[2], that means that each phone of the input sequence is pronounced during 5 to 13 frames in the output spectrogram. This duration varies depending on the type of phone (nasal and open vowels tend to be longer than close vowels for example [O'Shaughnessy, 1981]), but also depending on supra-segmental prosodic factors such as the applied style (e.g. utterances produced to convey thoughtfulness show prototypical elongations of ending syllables). Thus, this duration cannot be easily determined by rule, and needs to be inferred by the model from regularities learned during training. Learning this alignment from letters is even more challenging, because of the wider attention span needed to decode orthographic sequences (see Section 2.1.1).

To perform this alignment, the two families of TTS models studied in this thesis rely on opposing approaches. In Tacotron2 [Shen et al., 2018], an attention network [Bahdanau

[1] This procedure is called teacher-forcing training. It helps to achieve a faster convergence of the training.

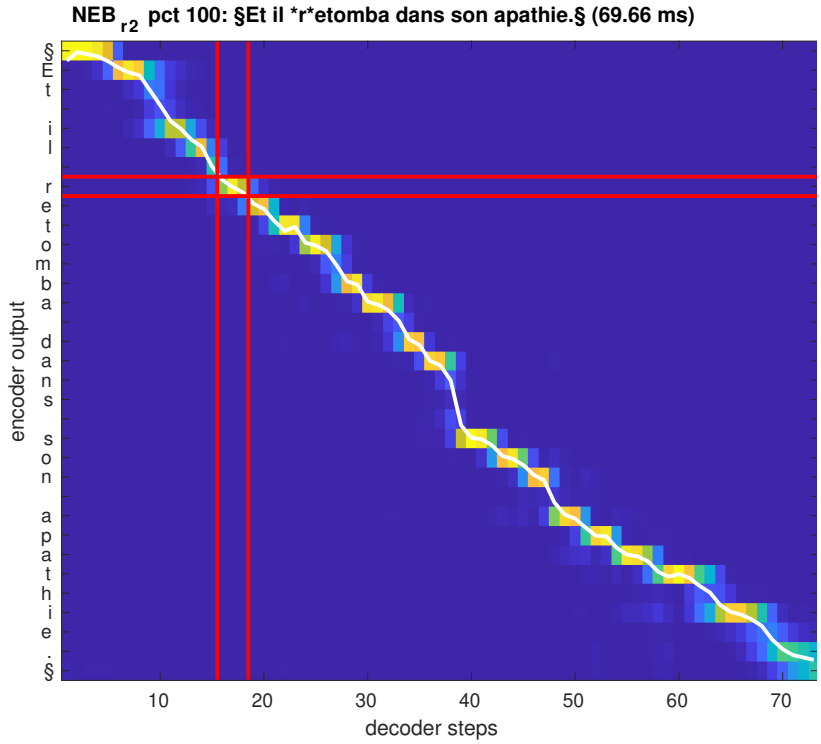
[2] The hop-size is usually 256, which corresponds to 11.61 ms by spectrogram frame with an audio sampling rate of 22.05 kHz.

et al., 2014] is implemented as the interface between the encoder and the decoder. For each spectrogram frame computed by the autoregressive decoder, this attention layer computes the relative weights given to each element of the input sequence. These weights, called **attention weights**, indicate the relative focus given by the model to the elements of the input sequence to predict the current spectrogram frame. This unsupervised alignment allows the training of Tacotron2 on any type of input symbols. On the contrary, FastSpeech2 requires the time-alignment of input sequences beforehand. Although computing this duration alignment is straightforward for phonetic inputs (all phones are pronounced in order in the audio output), it is unclear how to distribute the durations across the orthographic sequences. In order to train FastSpeech2 on orthographic inputs, we thus need an alignment mechanism to set the duration of input characters.

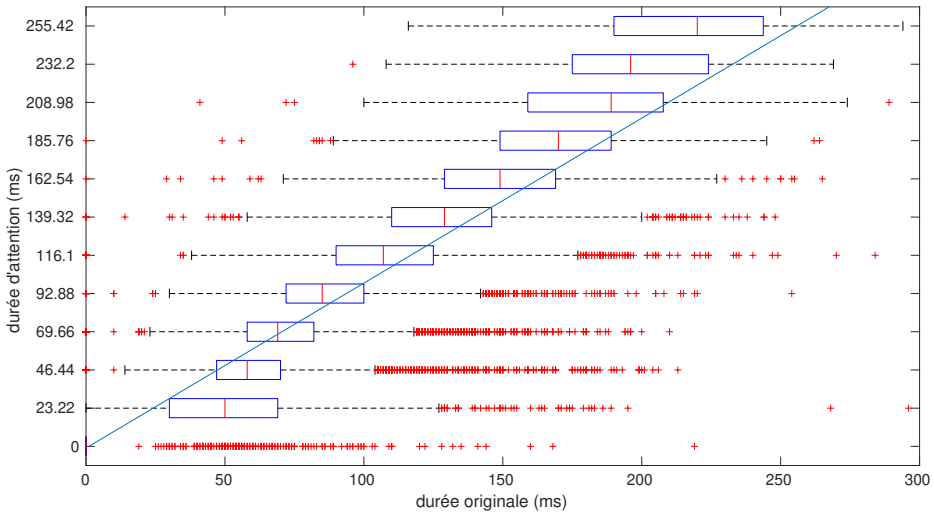
In this section, we demonstrate that the reading of attention maps computed by Tacotron2 can inform us about how orthographic sequences are processed by unconstrained alignment layers. The attention mechanism exhibits regularities that may indicate how neural TTS optimizes phonetic representations in order to produce the best audio quality, in particular with languages with opaque orthography like English or French. The observation of these regularities has enabled us to align the duration by phone with the corresponding characters. This alignment provides new perspectives for training duration predictors directly on letters, without the need for a Letter-to-Sound (L2S) front-end.

2.2.1 Preliminary Training of a Tacotron2 Model

In order to extract attention maps, we trained a Tacotron2 model following the configuration detailed in Section 2.1. This model is trained on a subset of our corpus described in Appendix A: only the speaker NEB is used in the experiment. The training follows the mixed-inputs procedure described in Appendix B. 5% of this corpus (2230 utterances) was randomly excluded as the test set. After 100 epochs, the attention maps computed from this test set are saved for further analysis. Two attention maps are computed by utterance: one in **inference mode** and one in **teacher-forcing mode**. The inference mode refers to the alignment path computed in absence of reference Ground-Truth spectrogram, which indicates the extrapolation capabilities of the model outside of its learning corpus. The teacher-forcing mode re-introduces target Ground-Truth spectrogram frames into the autoregressive process, which enforces the attention path to follow the same rhythm as the Ground-Truth. Only orthographic inputs are used in this section. Fig. 2.3a shows an example of such attention map computed in inference mode by the fully trained Tacotron2 from the input text: "§Et il retomba dans son apathie.§".



(a) Example of an attention map computed by Tacotron2 in inference mode. The white line shows the barycenter of attention weights.



(b) Correlation between phone duration and corresponding activation duration measured on the attention map.

Figure 2.3: Attention Map Analysis.

2.2.2 Automatic Segmentation of the Audio Signal

Despite the complex task performed by the attention layer, the general process can be illustrated by the example given in Fig. 2.3a. Once fully trained, the attention layer mostly produces monotonic paths at inference. The monotonicity is encouraged by the location sensitive mechanism [Chorowski et al., 2015], which indicates to the attention layer which characters of the input sequence have already been focused on during previous decoding steps. However, the monotonicity is not strictly constrained, which leads to two counterintuitive behaviors rarely observed, but worth mentioning:

1. Some quick **lookaheads**[3] may remain at word boundaries. We hypothesize that lookaheads are introduced to anticipate specific boundary patterns, like liaisons or pauses, which may not be fully defined by the contextualization performed by the text encoder.

2. **Residual focuses**[4] on the initial character are common during the synthesis of intra-utterance pauses. The initial character is systematically focused during the initial 130 ms of silence learned from the training dataset. When synthesizing intra-utterance pauses, the focus may switch to this initial character instead of the punctuation mark or space in which the pause appears. Intra-utterance pauses are very sporadic, which may make them unreliable for the attention mechanism, which instead generates pauses based on the utterance initial punctuation mark embedding.

Fig. 2.3a also highlights the skipping mechanism performed by the attention on orthographic input sequences. In the presented example, the number of characters focused at one point in the decoding process matches the number of phones in the audio output (pauses are considered as silence symbols in the phone output sequence). For instance, "n" and "s" are skipped in the word "dans", transcribed /d a$_{\sim}$/ in phonetic symbols. We hypothesize that the distribution of attention weights in time does indicate the frames during which the phones are pronounced in the output spectrogram. Thus, the skipping mechanism exhibited by this example distinguishes between character embeddings which have integrated a phonetic representation needed to produce the audio output, and other characters which have already fulfilled their goal of introducing contextual information in the input sequence.

In order to explore regularities of the attention maps, we proposed an automatic segmentation method to compute the duration of focus by symbol in the input sequence. This method takes advantage of the relative ease of interpretability of the single-head attention of Tacotron2. More complex attention mechanism like multi-head Transformers, also used in TTS architectures [Li et al., 2019], are inherently harder to apprehend, even though interfaces have been developed to allow human exploration of Transformer attention maps [Jaunet et al., 2021].

[3] Attention Lookahead refers to the attention pattern characterized by 1 or 2 frames of focus on one or several characters ahead, followed by a return to the character previously focused.

[4] Residual Focus describes the attention pattern characterized by multiple focuses on one character after the initial focus.

To compute frames corresponding to each input symbol, we apply the following procedure to every input symbol successively:

1. Check if this symbol reaches a minimum attention weight of 0.35 at any time in the attention map. This threshold excludes symbols that are never the main focus of the attention. These characters are called "mute" in the following.

2. For the frame on which the attention weight is maximum, check that the current character has the maximum attention weight among all characters of the input sequence. This avoids the prediction of two characters being generated in the same frame. If the maximum is not on the current character, it is considered mute too.

3. If both conditions are met, the current character is considered to be generated during this frame, as well as during adjacent frames in which the attention weights are maximum on this character. This frame selection inherently ignores lookahead and residual attention focus described above, since only the most prominent group of adjacent frames is considered to compute the duration of one character. This limitation is discussed in Section 4.3.

We evaluated this segmentation procedure in [Lenglet et al., 2022a]. To do so, we extracted the duration from teacher-forcing attention maps with the presented method. We then compared these durations to the Ground-Truth phone segmentation obtained by semi-automatic alignment. The teacher-forcing mode ensures the synthetic model follows the same dynamic as the Ground-Truth. Results are given in Fig. 2.3b. We measured a Pearson correlation coefficient of 0.88 between predicted and real durations (punctuation marks and spaces excluded). This high correlation indicates that the proposed segmentation method is fitted to automatically analyze duration of phones pronounced in synthesis from the attention map. This method is used in the following as a post-hoc duration predictor for Tacotron2.

2.2.3 Identification of Activation Patterns

The automatic segmentation procedure described in Section 2.2.2 allowed us to examine the regularities of attention activation patterns. In previous work, Perquin et al. [2020] have already established that the encoder of a neural TTS model such as Tacotron [Y. Wang et al., 2017] computes phonetic representations from orthographic sequences given as inputs, even when trained exclusively on orthographic sequences. However, phonetic transcription from orthographic characters is not a one-to-one mapping. On average, 2.3 letters are needed to express one phone in opaque languages like French and English [Bosse & Valdois, 2009]. The distribution of model's attention between involved characters remains unclear in [Perquin et al., 2020]. Our goal here is to understand how phonetic representations emerge from orthographic sequences, particularly in the case of phones written with multiple characters, called **complex phones** in the following.

To evaluate how attention is distributed in complex phones, we scann the attention maps from our test corpus produced in inference mode with Tacotron2. The 12 most common cases of complex phones in French are selected for this analysis. The activation duration by character is given in Fig. 2.4 for this selection.

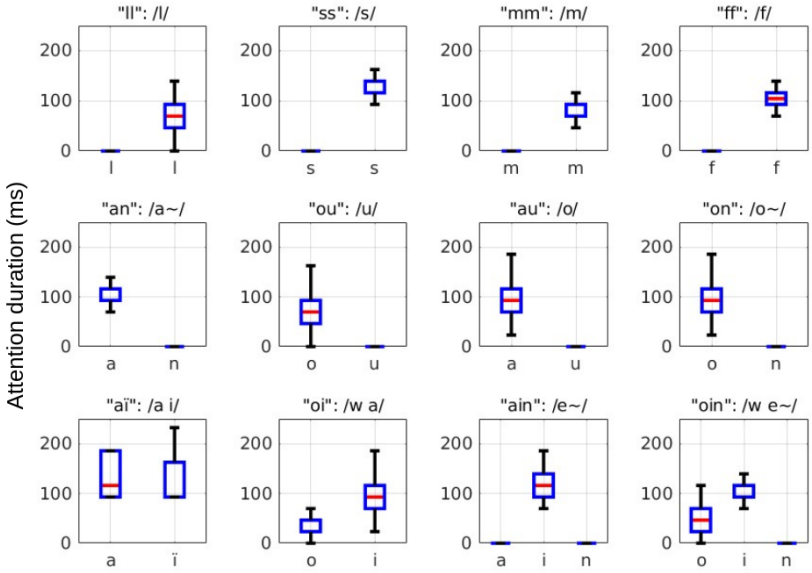


Figure 2.4: Distributions of durations of activation (ms) of character sequences: when one phone is encoded by two letters, the second character gets mostly activated in context of double consonant letters, while the first is activated in context of vowel letters.

Table 2.1: Activation rules on recurrent letter patterns. C and V stand for consonant and vowel letters respectively. "_" stands for the mute phone.

Letters	Phones	Examples
C C	_ C	"a**nn**ée", "e**ll**e", "au**ss**i"
V V	V _	"t**an**te", "**ou**", "**au**"
V V V	_ V _	"**eau**", "p**ain**"
"h"	_	"**h**aut"
"ch"	s^ _	"**ch**ateau"

Following the examples of Fig. 2.3a, only one character input stands out from the group of characters involved in each phone. We empirically deduced a set of rules from these observations, summed up in Table 2.1. We hypothesize that in the case of complex phones, some characters only serve as contextual information to enrich the most relevant character of the orthographic sequence. As a result, only this character whose embedding encodes duration and spectral cues of the phone to synthesize is relevant to go through the decoder. As a general observation, contextual characters seem more likely to enrich the previous character, as shown by the double vowels scenario. This logic also applies to double consonants, since the first one can contribute to change the previous phone (usually a vowel): "elle" is transcribed /e _ l _/ whereas "e" alone is generally pronounced /x^/. In triple vowels "eau", "ain" and "ein" (resp. /o/, /e~/ and /e~/), the second and third characters already encode the corresponding phone, since "au" and "in" are respectively pronounced /o/ and /e~/. This makes this easier for the model to learn to systematically mute the first character.

These L2S transcription rules, based on statistical learning, is to our knowledge the first attempt to described the distribution of TTS models' attention when dealing with opaque languages. Thus, the proposed method eases the analysis of orthographic embeddings, since character embeddings can also be labeled with their phone counterpart. Additionally, this one-to-one L2S mapping also opens the route toward training duration prediction tasks directly on orthographic input, by providing duration alignment rules for letter inputs. This method allowed us to train FastSpeech2 on mixed-representations without the need for a L2S front-end, as described in Section 2.3. Finally, this mapping was also used to set the targets of the phone prediction task implemented at the output of the text encoder of Tacotron2 and FastSpeech2 in following experiments.

2.3 Training FastSpeech2 with Orthographic Representations

As described in Section 1.1.4.2, FastSpeech2 was initially designed for phonetic inputs. FastSpeech2 is not an isolated case; indeed, modeling phonetic sequences from orthographic inputs is not a trivial challenge. Neural end-to-end approaches initially found little success on this task: J. Taylor and Richmond [2019] reported L2S error rates close to 10% in 2019. These results have encouraged most researchers in the field to primarily focus their TTS models on phonetic inputs, and to rely on a separate front-end to convert texts to phonetic sequences: for the Blizzard Challenge 2023 [Perrotin et al., 2023], 9 teams out of 18 used eSpeak as front-end phonetizer [Dunn & Vitolins, 2019]. Although this method performs well in most cases, it generally learns a normative version of pronunciation rules, which gets rid of the idiosyncratic variability. However, speakers tend to produce a wide range of phonetic variants from the same textual content, mainly due to sociodemographic factors. As an example, regions of origin [Resnick, 2012], social classes [Stuart-Smith et al., 2014] and ages [Foulkes & Docherty, 2006] have been shown to impact phonetic productions. Conditional Random Fields (CRF) have been proposed to adapt L2S front-end with speaker-idiosyncrasies [Tahon et al., 2016]. However, the proposed method avoids the L2S front-end entirely.

As explained in Section 1.1.1, this thesis advocates that orthographic inputs should be the normative symbols that should constitute the basis of TTS representations. However, the limitations of orthographic sequences described above should be taken into consideration in order to maximize performance of neural models when using text input. Thus, this section describes the adaptations we have implemented in FastSpeech2 in order to maximize performance on orthographic inputs.

2.3.1 Model Adaptations

This section describes the modifications implemented on the original FastSpeech2 model described in Section 1.1.4.2. The modified model is illustrated in Fig. 2.5. The exhaustive list of hyperparameters and the default training procedure is detailed in Appendix C.

Fig. 2.5 illustrates the modifications implemented on the original architecture. Following early implementations of FastSpeech2, the pitch predictor is trained on fundamental frequency values in semitones, instead of continuous wavelet transforms [Vainio et al., 2013] in later work. Pitch and energy values are extracted using WORLD pre-processing toolbox [Morise et al., 2009], and are averaged by phone, and normalized by speaker. Pitch and energy are thus predicted by phone instead of frame, and the corresponding embedding is also added at the phone-level before the length regulator. The energy predictor is cascaded with the pitch predictor (see Fig. 2.5). A Tacotron2-like postnet is added after the mel-spectrogram prediction. This postnet models finer-grained temporal patterns through a stack of convolutional layers. The spectral residual computed by the postnet is added to the prediction of the decoder. The postnet is trained with MAE spectral reconstruction loss after the addition of the residual, following formula 2.2.

Duration by phone is computed as $log(1 + phon_{dur})$, with $phon_{dur}$ the number of frames during which this phone is pronounced in the target mel-spectrogram. This arbitrary addi-

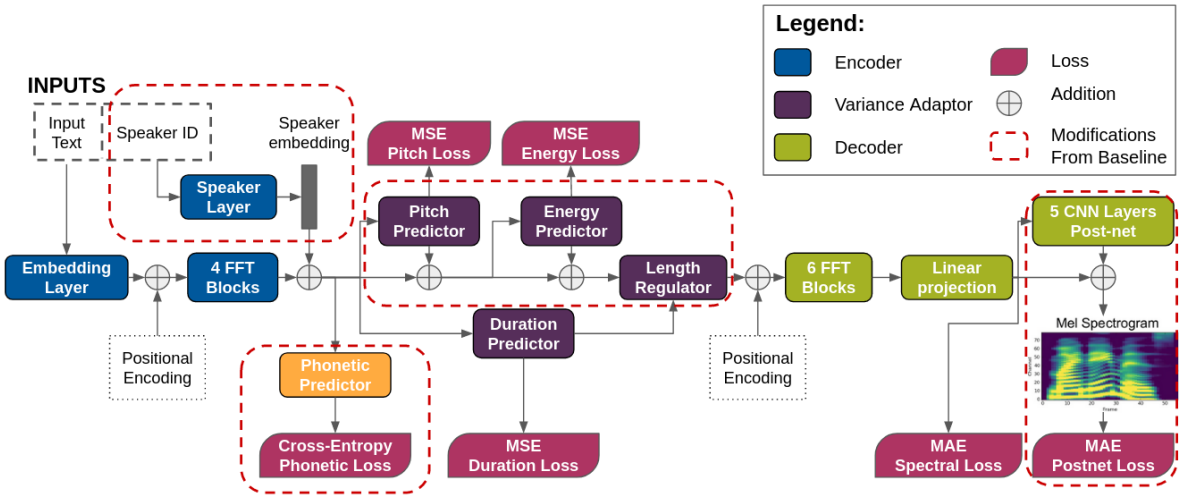


Figure 2.5: Modified FastSpeech2 Architecture

tion in the logarithm allows for zero duration to be predicted correctly by the model. This adaptation is necessary in the case of orthographic inputs.

$$L_{FS} = L_S + L_{PS} + L_{dur} + L_p + L_e \tag{2.2}$$

with L_{FS} the total loss of FastSpeech2, L_S the MAE spectral loss, L_{PS} the MAE spectral loss after the Postnet, L_{dur} the MSE duration loss, L_p the MSE pitch loss and L_e the MSE energy loss.

The implementation is modified to admit letters as inputs. When using letters, phone durations are attributed following the rules set in Section 2.2.3. This enables to train FastSpeech2 directly on orthographic sequence, and use orthographic sequences at inference. Thus, it allows the model to handle idiosyncrasies on its own, which can otherwise be an issue with the L2S front-end [Bailly et al., 2023].

2.3.2 Enhancement of FastSpeech2 with a Phonetic Prediction Task

The performance of the mixed-input framework highlights neural TTS capacities to set a shared latent space for both orthographic and phonetic representations. However, orthographic inputs still suffer from the relatively poor coverage of the training corpus compared to the potential variability of French. Depending on the context, orthographic characters can be pronounced more than 7 different ways (e.g. "e" as [ã] in "rend", [a] in "femme", [ɛ] in "lemme", [ø] in "feu", [œ] in "peur", [œ̃] in "agenda", [ɛ̃] in "bien"). Extracting regularities from the corpus is challenging for neural models, in particular with some occurrences being so rare in the corpus. Heterophonic homograph[5] disambiguation, for example, suffers from the unbalanced distribution of homographs pairs in most training corpora, resulting in the most common variant being systematically produced by TTS. This phenomenon is detrimental to wider usage of TTS since homographs convey different meanings and thus are very confusing when mispronounced.

In order to help the TTS model to learn regularities from orthographic sequences and further disambiguate homographs, we proposed the introduction of a phonetic prediction layer in the classical encoder-decoder pipeline of neural TTS. This neural layer is introduced before the decoder, as illustrated in Fig. 2.5. It is composed of a fully connected layer followed by a softmax function. This prediction layer computes a probability distribution over the 38 possible phone output symbols for each element of the input sequence (phonetic symbols are fully described in Appendix A). The prediction is performed after the addition with the speaker embedding bias to account for idiosyncrasies. This predictive layer is trained on the one-to-one L2S mapping proposed in Section 2.2. This secondary categorization task is trained with a cross-entropy loss, following equation 2.3. This additional loss is back-propagated through the text encoder and is trained along the initial model pipeline.

[5] Two or more words spelled alike but different in pronunciation.

$$L_{FS} = L_S + L_{PS} + L_{dur} + L_p + L_e + L_{phon} \tag{2.3}$$

with L_{FS} the total loss of FastSpeech2, L_S the MAE spectral loss, L_{PS} the MAE spectral loss after the Postnet, L_{dur} the MSE duration loss, L_p the MSE pitch loss, L_e the MSE energy loss, and L_{phon} the cross-entropy phonetic loss.

While explicitly providing the model with additional regularities to help its training process, this secondary task enables the training of the text encoder without the need for audio recordings. Indeed, the phonetic prediction layer uses the output of the text encoder as input. Because the cross-entropy phonetic loss is back-propagated through the text encoder, the encoder itself and the input embeddings can be trained solely on <orthography|phonetic> pairs. This enables the training of models on rare contexts and on updated vocabulary, by augmenting the usual audiobooks corpus with various online resources, without the need for corresponding audio recordings. Online dictionaries like Robert provide a wide variety of rare words in context, as well as homographs, that can be included with this method to enrich the capacities of the text encoder.

Since this phonetic prediction task constrains the latent space used as input by the acoustic decoder, we hypothesize that this additional training may help the text encoder to compute meaningful acoustic representations on a wider variety of words and contexts. This hypothesis was validated by our contribution to the Blizzard Challenge 2023, which featured this phonetically-enhanced FastSpeech2 [Lenglet et al., 2023c]. The benefits of this phonetic prediction task are reported in Section 2.4.

2.4 Evaluation of the FastSpeech2 Baseline (Blizzard Challenge 2023)

We decided to compete in the Blizzard Challenge 2023[6] with a FastSpeech2 model enhanced with the mixed inputs training and the phonetic prediction layer. The Blizzard Challenge 2023 featured two main tasks [Perrotin et al., 2023]:

- The **Hub-task** evaluated the models on a large corpus of more than 50 hours of audiobooks recordings. The speech quality was evaluated with Mean Opinion Scores (MOS). Intelligibility was evaluated on heterophonic homographs disambiguation and Semantically Unpredictable Sentences (SUS).

- The **Spoke-task** evaluated the transfer learning ability of the model to another speaker with a limited corpus of 2 hours.

[6]To clarify the sequence of events, although our Fastspeech2 model was trained at the start of my PhD in 2020, I did not evaluate this model in comparison with the baseline version of Fastspeech2 until the Blizzard Challenge 2023.

We entered both tasks with the same multi-speaker model. This section describes the training of our model and discusses our model's performance in comparison with the FastSpeech2 baseline trained by the organizers. This comparison highlights the benefits of our proposed training using orthographic inputs compared to traditional phonetic inputs paired with an L2S front-end.

2.4.1 Model Training

The FastSpeech2 architecture used in this experiment follows the implementation described in Section 2.3. This model is trained on a subset of the corpus shared by the organizers: only the utterances with phonetic alignments were considered in this training. We added two additional speakers to the Blizzard dataset. Following Blizzard rules for the challenge, the two additional speakers are taken from open-access online databases. Additionally, the phonetic prediction layer is used to further train the text encoder on non-audio data, extracted from the online dictionary Robert and various online resources, explained in Hajj et al. [2022]. Our training dataset is specified in Table 2.2.

Table 2.2: Multi-Speaker Training Dataset for the Blizzard Challenge. Durations are given in hh:mm:ss.

Speaker	**Metadata**		**Audio**	
	Dataset	**Gender**	**Duration**	**# Utt**
NEB	Blizzard	Female	33:33:41	44 029
AD	Blizzard	Female	2:04:53	2 515
DG	LibriVox [Kearns, 2014]	Male	6:17:22	7 539
RO	SIWIS [Honnet et al., 2017]	Female	0:35:21	586
Dictionary	Robert	-	-	95 879
Homographs	Various [Hajj et al., 2022]	-	-	17 285
Total	**-**	**-**	**42:31:17**	**167 833**

2.4.1.1 Multi-speaker adaptations

First, we opted for trainable speaker embeddings. When training on multiple speakers, we train a set of embeddings that are added to all the character/phone embeddings of the sequence outputted by the text encoder, as illustrated in Fig. 2.5. These embeddings are trained alongside the rest of the model through the same loss function. This procedure requires speaker labels for the entire corpus, and limits the inference to the speakers seen during training. Speaker embeddings can be seen as an offset in the acoustic latent space computed by the text encoder. This offset encodes specific speaker features, like pitch and speaking rate, but also sociophonetic attributes as demonstrated by Bailly et al. [2023].

Prosodic predictions in FastSpeech2 are re-injected into the model by the addition of pitch and energy embeddings. These embeddings cover the range of prosodic values seen in the corpus. In case of multi-speaker training, this range will be unequally distributed: male and female speakers typically cover separated ranges of pitch with few overlapping. Using non-normalized values then leads to some prosodic embeddings being never used. In order to ensure a normal distribution of values across the range, pitch and energy are normalized by speaker. The feature range then matches the maximum variation produced by one of the speaker relative to his/her mean.

Finally, phonetically-aligned dictionary inputs are duplicated for each speaker. The phonetic prediction layer is implemented after the addition of the speaker embedding, which makes this prediction speaker-dependent. Non-audio inputs are necessarily aligned on the normative phonetic transcriptions of the given textual content. Nonetheless, this duplication of the non-audio maximizes the variety of contexts seen by the model for each speaker.

2.4.1.2 Training Procedure

The training process follows the multi-speaker procedure described in Appendix C. The model is initially trained for 100 epochs on NEB. Then the model is trained on the full multi-speaker dataset for 50 epochs. Finally, the model is fine-tuned for 50 epochs on an evenly distributed corpus across speakers. We used Waveglow as a vocoder (see Appendix D.2).

2.4.2 Performances of the Phonetic Prediction

The overall accuracy of the phonetic prediction was evaluated as a preliminary study for the Blizzard Challenge. As a test set, we randomly extracted 2230 additional utterances recorded by the same NEB speaker from the original M-AILABS corpus [Solak, 2019]. These utterances are not part of the dataset shared by the Blizzard organizers, thus they have not been seen by the model during the training phase. This test set is synthesized twice, using both orthographic and phonetic inputs. Phonetic prediction by input characters are given as confusion matrices in Fig. 2.6.

For the 108 168 orthographic characters of this test set, the overall accuracy reaches 0.984 (0.997 when excluding muted characters and pauses). Interestingly, most remaining errors (reported in Table 2.3) are mispredictions of schwas or confusions between close phonetic variants due to mispredictions of vowel harmony[7]. Most errors with muted characters are miss-predicted liaisons on ending /r/, /t/ or /z/. Note that the errors highlighted here may just reflect divergences between the ground truth and the model decision on optional liaisons. On the other hand, when using phonetic inputs, this prediction is almost flawless, reaching 0.993 overall, and 1.00 when excluding pauses.

[7]Vowel Harmony is the optional adaptations of vowels within a word in order for all vowels to share certain phonological features: frontness or backness, rounding, nasality, etc. For example: "J'ôte" [o] VS "Nous ôtons" [ɔ]

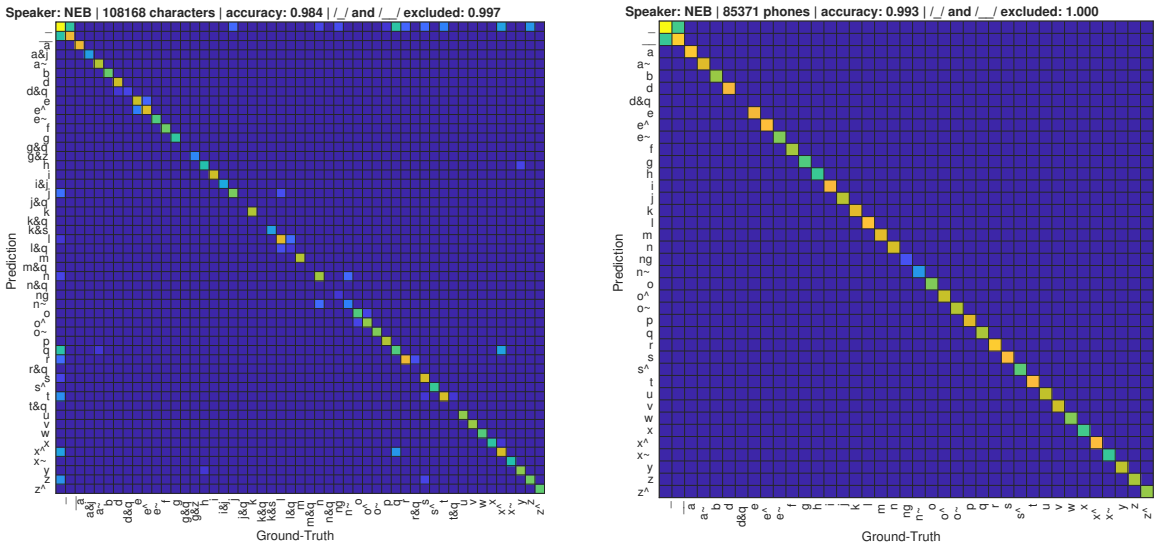


Figure 2.6: Confusion Matrices of the phonetic prediction layer, for orthographic inputs (left) and phonetic inputs (right).

Confused Phones		Typology	Example
/q/	/x^/	Schwas VS inserted vowel	"quelqu**e**s rares fenêtres"
/o/	/o^/	Optional Vowel Harmonic choices	"**Ô**tons nos souliers"
/r/, /s/, /z/, /t/	/_/	Optional Liaisons	"si tu n'es pas_heureux"

Table 2.3: Most common confused phones in Fig. 2.6.

To our knowledge, this method outperforms other classical L2S methods found in the literature [Yoon et al., 2023]. Using similar TTS pre-trained representations, Perquin et al. [2020] reported a phone error rate of 12.8% for a similar L2S task on the first version of Tacotron [Y. Wang et al., 2017]. Although our reported performances are very promising, the results of our phonetic prediction does not guarantee that the corresponding synthesis perceptually matches the predicted sequence of phones. As a result, the 10% L2S error gap reported by J. Taylor and Richmond [2019] may not be fully solved by this method. However, this automatic L2S transcription method through internal representations of neural TTS is very reliable. We have used this method to generate new phonetic alignments to enrich our corpus, with very few corrections needed. Additionally, this method provides a speaker-dependent phonetic prediction, by taking into account the sociophonetic habits of the speaker [Bailly et al., 2023].

2.4.3 Disambiguation of Homographs

Intelligibility assessment on heterophonic homographs evaluated by the Blizzard organizers is reported in Fig. 2.7. Our model **N** achieves an average score among all systems. Our model shows global improvements over the FastSpeech2 baseline (annotated **BF**). **BF** is trained on phonetic inputs only, and relies on the eSpeak[8] front-end for the G2P transcription.

[8]https://espeak.sourceforge.net/

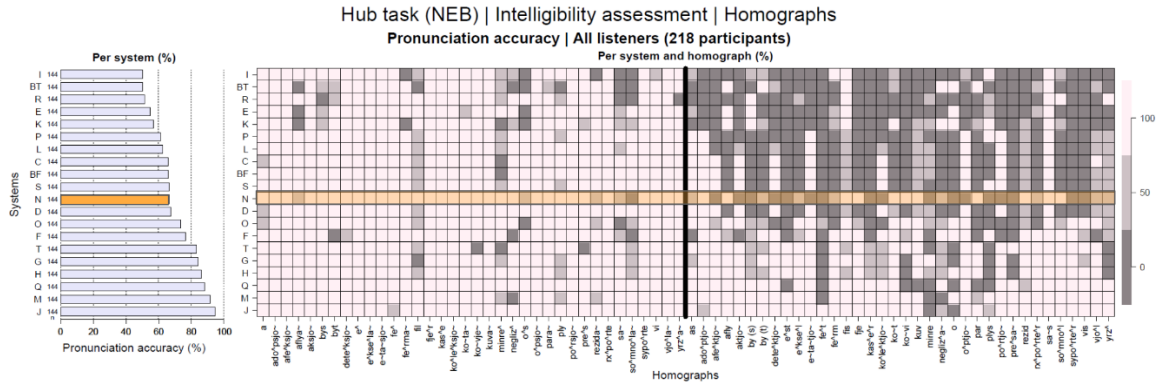


Figure 2.7: Homograph intelligibility scores for the Hub Task. Our model N is highlighted in orange. The left graph shows the percentage of correct pronunciation by system. The right graph shows this intelligibility assessment by homograph.

Table 2.4: Examples of French homographs. Disambiguation is validated (✓) if the pronunciation accuracy is 100% for both variants.

Homograph	Variant A			Variant B			# Examples in Corpus	Disambiguation
	Phonetic	POS	English	Phonetic	POS	English		
Fier	f j e^r	Adj	proud	f j e	Verb	trust	366	✓
Fils	f i s	Noun	son	f i l	Noun	wire	261	✓
Convient	k o~v i e~	Verb	suit	k o~v i	Verb	invite	181	✓
Portions	p o^r t j o~	Verb	carried	p o^r s j o~	Noun	servings	145	✗
Intentions	e~t a~s j o~	Noun	intents	e~t a~t j o~	Verb	initiated	141	✗
Options	o^p s j o~	Noun	options	o^p t j o~	Verb	opted	117	✗

More specifically, our model performs very well on homographs that have been seen with enough examples in its homograph corpus, as illustrated by Table 2.4. “Fils” (261 examples) has an intelligibility score or 100% for both variants, whereas systems with overall better scores do not achieve such accuracy on this specific homograph. This is also true for “convient” (181 examples) or “fier” (366 examples), with the most common forms /kɔ̃vjɛ̃/ and /fjɛʁ/ being systemically pronounced by other TTS regardless of the context. On the contrary, “options” (117 examples), “intentions” (141 examples) and “portions” (145 examples) also appear in the homograph training corpus, but with fewer examples. The number of examples and the balance between variants impact the performance of the system. However, the proposed method helps to disambiguate homographs if enough examples are given during training. From empirical observations, at least 150 examples seem to be needed in order to achieve robust disambiguation.

2.4.4 Speaker Adaptation

The multi-speaker performance of our model are evaluated by the Spoke Task. Mean Opinion Scores of the quality assessment are reported in Fig 2.8. Our model **N** showed the same performances than the FastSpeech2 baseline **BF**. **BF** employed a similar trainable speaker

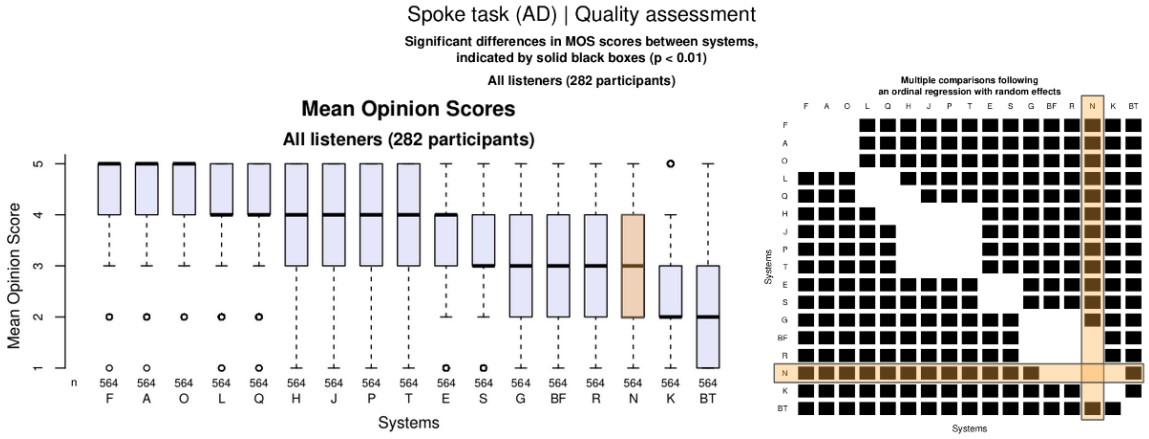


Figure 2.8: Quality Assessment on the AD voice evaluated in the Spoke Task. Our model N is highlighted in orange. The right graph shows MOS distribution by system. In the right graph, black squares show that the difference between the two models is significant ($p < 0.01$).

embeddings than our proposed model. Neither the addition of two additional speakers in the dataset nor the training on phonetic representations seemed to improve the multi-speaker performance. As mentioned in Appendix D.2, our Waveglow vocoder has only been trained on NEB data. This lack of multi-speaker representations has probably negatively impacted the extrapolation performances of our vocoder.

2.4.5 Discussion: Mixed Inputs combine the Best of Both Worlds

The presented results show the benefits of combining both orthographic and phonetic representations as inputs of neural TTS models. Unlike poor performance stated by J. Taylor and Richmond [2019], TTS models trained on mixed input are able to combine the versatility of orthographic inputs with the accuracy of phonetic representations. To achieve this phonetic accuracy, a particular attention is required regarding the training procedure of such neural models.

The proposed addition of a phonetic prediction layer at the output of the text encoder provided further improvements of the overall phonetic accuracy of the model. Thanks to the specific training of the text encoder on multiple external text resources, the phonetically-enhanced FastSpeech2 was able to disambiguate heterophonic homographs better than most other models which rely on a classical L2S front-end. Moreover, by taking into account the speaker variability, the proposed phonetic prediction layer is able to learn speaker sociophonetic behaviors [Bailly et al., 2023]. Thus, the predicted phonetic sequence integrates schwas, liaisons and vowel harmony relative to the speaker's idiosyncrasies, which is generally ignored by classical L2S front-ends.

More careful analysis of L2S front-end systems used by Blizzard participants revealed that the models which had the best pronunciation accuracy integrate pre-trained representations

from Large-Language-Models (LLM) like BERT [Devlin et al., 2018][9]. The benefits of using LLMs for homograph disambiguation is well established. Hajj et al. [2022] compared the performances of Part-Of-Speech (POS) tags, LLM representations and phonetic predictions from Tacotron2 representations. The authors found similar results, with LLMs outperforming other evaluated methods. Although these LLMs provide very promising results for various speech-related tasks, they should not be seen as a perfect solution. LLMs require a large memory footprint, which limits their potential for low resource devices. Additionally, they require extensive datasets and high computation power during training, which make them unaccessible to train for most people. As a result, pre-trained LLMs are generally used, with little to no information regarding their training procedure, which can lead to various biases in the evaluation of sub-tasks.

2.5 Discussion: Establishment of the TTS Baseline

The combined training of orthographic and phonetic representations relies on the ability of the model to learn regularities in the L2S mapping. This mapping is not trivial, as demonstrated by the mitigated performances of standard neural L2S front-ends. In the presented work, we hypothesized that the best performance should be provided by an alignment method which relies on the understanding of internal representations of neural models. The analysis of internal representations, and in particular the tracking of acoustic and phonetic features encoded in the internal layers of neural models, will be discussed in Chapter 4. We will demonstrate how the L2S alignment proposed in this chapter benefits the acoustic analysis of text embeddings.

We detailed the modifications implemented in our two models, Tacotron2 and FastSpeech2. We have shown the benefits of training neural TTS on both orthographic and phonetic input. The proposed prediction of phonetic symbols from internal orthographic representations not only outperformed previous G2P approaches, but also showed promising performances in heterophonic homographs disambiguation and phonological variation. This method was evaluated on FastSpeech2, but could be identically applied to any TTS models. We also applied this phonetic prediction layer to Tacotron2 in later works.

This chapter described the design process of our two French TTS baselines used as the basis of all following work in the thesis. The two models were selected as representatives of two families of state-of-the-art architectures: Tacotron2 (recurrent TTS) and FastSpeech2 (parallel Transformer-based TTS). These models embody two distinct ways to apprehend voice synthesis. On one side, Tacotron2 favors weakly supervised internal representations. The spectral reconstruction loss is supposedly taken as the sole goal of the neural model, which should build the best internal representations possible to maximize its accuracy. On the other hand, FastSpeech2 is designed around the prediction of explicit low-level prosodic features, i.e. phone durations, energy and F0. The model is specifically designed to predict these features at specific locations, which inherently constrains its internal representations. This method supposes that important acoustic features are known in advance, and emphasizes the controllability and the robustness of these features at inference time.

[9]Models **O** to **J** (top 8 performances) use LLMs pre-trained representations [Perrotin et al., 2023]

Although the robustness of the prediction of prosodic features is crucial for real-life applications of TTS systems, only a deeper understanding of internal learning mechanisms of neural models can provide interesting insights on how to push further the already good performances of such architectures. The finer understanding of the tasks performed by each layer of neural models may help designing more thoughtful architectures, as has already been the case with visual recognition networks [Zeiler & Fergus, 2014]. Therefore, we have considered both Tacotron2 and FastSpeech2 for most of the experiments presented in next chapters, instead of opting for one single model.

Chapter 3

Linguistic Prosody Modeling through Input Sequences

Contents

Chapter Highlights

This chapter presents our works to better control **Linguistic Prosody** in Neural TTS. For this purpose, we studied the impact of how input data are presented to the model. Two main axes are explored: 1) We studied the **effects of the corpus segmentation**. We showed that the average duration of utterances impacted both the phrasing and the spectral accuracy, in opposite directions. 2) We introduced **initial punctuation marks** as a way to model inter-utterances prosodic patterns. For these two studies, we showed how the **multi-dimensional analysis** of listening test results provides valuable insights about the underlying factors of perceptual judgments.

Related contributions: [Lenglet et al., 2021]

Despite the engaging performance reported by state-of-the art neural TTS, training on isolated utterances inherently limits model capacity to predict consistent prosodic patterns. Natural linguistic prosody not only arises from the syntactic structure of the current utterance [Liberman & Prince, 1977], but also from the speaker's understanding of the wider discourse he/she is conveying. The generation of long forms of speech is therefore conditioned by the TTS ability either to learn how to produce multiple utterances at once, or to convey enough contextual information between successive smaller chunks of speech. However, learning to generate longer form of speech by TTS has been shown to be challenging for multiple reasons:

1. Large batch size including long utterances demands high computation memory.
2. Learning long-term dependencies is a challenging task for recurrent models [Hochreiter et al., 2001]. Transformer-based architectures may alleviate this issue [Vaswani et al., 2017], but at the cost of losing the causal relationship in time sequences [Shen et al., 2020].
3. Style control, which is the ultimate goal of this PhD thesis, generally uses utterance-level style embeddings [Y. Wang et al., 2018; Y.-J. Zhang et al., 2019], which means that the shorter the utterances, the finer it is possible to tag speech styles.

These reasons oriented our work towards the generation of shorter speech segments, with an increased focus on contextual modeling. The use of audiobooks to train TTS should be an opportunity in that regard, since the inherently sequential nature of the corpus provides direct access to the causal relationship between successive utterances. Indeed, during recording sessions, the voice actor reads several paragraphs in one go, resulting in prosodic patterns that expand wider than the scope of individual utterances. However, the sequential nature of the corpus is generally ignored. During training, the corpus is divided into smaller batches, within which each utterance is considered as an individual sample. The model task during training then consists in predicting the output spectrogram as close to the original recording as possible. This training process cannot assess the underlying information structure of the training corpus (if any).

On the other hand, conditioning the prediction process of neural TTS on contextual information has shown great potential in the literature. Oplustil-Gallegos et al. [2021] proposed to transmit linguistic context between successive utterances. The authors compared the benefits of the combination of the current text with textual and/or acoustic representations from the previous utterance, at word and utterance-level. They found the best results with the mixed combination of utterance-level acoustic features and word-level textual representations. Similarly, Pascual et al. [2019] proposed to initialize the hidden states of the recurrent decoder of their MUSA model[1] with the final state of the previous decoded utterance. This method follows the stateful paradigm which also found success in other speech-related tasks like turn prediction [Ji et al., 2016; B. Liu & Lane, 2017]. These results advocate for the benefits of contextual information to model linguistic prosody in successive utterances.

[1] https://github.com/santipdp/musa_tts

However, the previously-mentioned methods allow little control from the operator during inference. Hidden states of recurrent networks cannot be chosen by rule. Therefore, in both cases, the preceding utterance fully set the contextual bias to apply. On the contrary, we envision a bias method through input sequences in order to extend our expressive control. In Section 3.1, we present our proposed segmentation and annotation of input data in order to maximize our TTS performance on shorter sequences. We thus introduce the proposed **linking punctuation marks** as a way to convey contextual information through text inputs. The segmentation of the corpus is evaluated in Section 3.2, followed by the evaluation of the control provided by the linking punctuation in Section 3.3.

3.1 Segmentation and Annotation of the Corpus

The common approach to neural TTS evaluation, seen in events like the Blizzard Challenge [Perrotin et al., 2023; Zhou et al., 2020], is to compare multiple models on the same corpus to evaluate the resulting synthesis quality. This process minimizes the importance of input data structuring, which ultimately shapes the output of any deep learning model. One complementary work is to evaluate multiple segmentations of data structuring on the same TTS model. This section describes this approach.

3.1.1 New Segmentation in Shorter Utterances

Publicly available corpora designed to train TTS, like M-AILABS [Solak, 2019], LJSpeech [Ito & Johnson, 2017] or SIWIS [Honnet et al., 2017] are generally composed of audiobook extracts read by one or more speakers, segmented in thousands of utterances. The utterance segmentation is generally automatically generated from silences detected in the recordings, without further explanations. As a result, utterance boundaries often match sentences, but not always. Automatic segmentation therefore may not produce syntactically consistent utterances, ultimately resulting in too much prosodic inter-utterance variability which degrades the model learning performance.

The M-AILABS French dataset [Solak, 2019] was used as a starting point for this work. This corpus includes more than 190 h of recorded speech, segmented in utterances from 1 s to 20 s, given with corresponding orthographic transcripts. Recordings come from the free public domain audiobooks LibriVox database [Kearns, 2014]. We selected a subset of the recordings made by Nadine Eckert-Boulet (NEB), for a total duration of about 33 h. Each book duration and corresponding number of utterances are given in Table 3.1. Audio files were originally sampled at 44.1 kHz, but we re-sampled them at 22.05 kHz.

For the reasons explained in this chapter's introduction, the M-AILABS segmentation may not be suited for TTS training: the segmentation is approximate, with a median duration of 6.44 s. Moreover, in the original clips shared by M-AILABS[2], recordings are bounded with 500 ms of silence (zeros in the waveform). These silences do not correspond to the recordings, but have been artificially added to each audio clip after segmentation.

[2]https://www.caito.de/2019/01/03/the-m-ailabs-speech-dataset/

Table 3.1: Duration and number of utterances in each book used from LibriVox used in this section.[3]

Book	Duration (hh:mm:ss)	# Utt Original Seg	# Utt New Seg
Les Mystères de Paris	21:33:28	12 285	28 333
Mme Bovary	11:07:25	5 775	14 417
Total	**32:40:53**	**18 060**	**42 750**

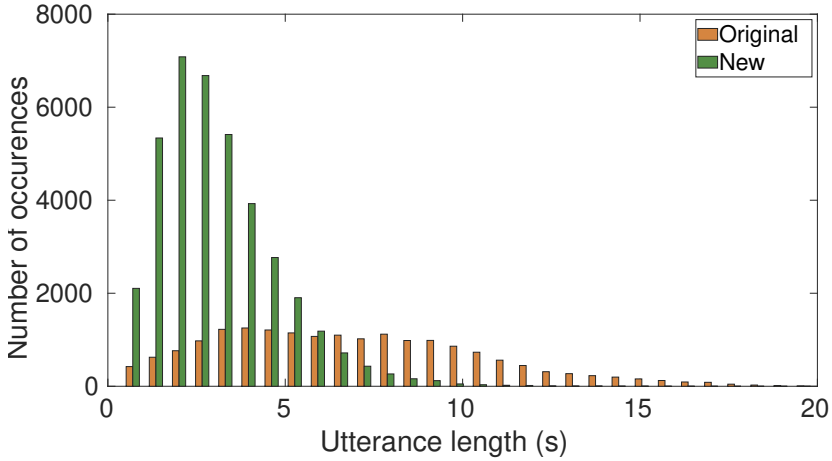


Figure 3.1: Distribution of utterances length of original and new segmentation. Source: [Lenglet et al., 2021].

In order to reduce this average duration, we went back to the original recordings of LibriVox [Kearns, 2014] and segmented chapters based on silences of at least 400 ms. This duration usually corresponds to pauses made between speaking turns in conversations [Bailly & Gouvernayre, 2012]. 94.56% of silences coincide with punctuation marks in this corpus. For the others, a comma is added by default at the end of the utterance. Timestamps were hand-checked for each utterance to ensure optimal segmentation[4]. Additionally, silence boundaries were replaced by 130 ms of room tone extracted from the original recordings (thus including breath noises, lips smacks ... if any). This duration matches the initial and final silence durations found in other speech databases such as LJSpeech [Ito & Johnson, 2017].

Table 3.1 shows duration and number of utterances of the obtained segmentation and Fig. 3.1 gives the distribution of utterances length of the original and the proposed segmentation. Median utterance duration (resp. first and third quartiles) is reduced from 6.44 s (3.88 s and 9.26 s) to 2.77 s (1.89 s and 3.95 s). 82.5% of utterances of the new segmentation last between 1 s and 5 s, and 0.25% of utterances last more than 10 s. 1336 utterances are unchanged, which corresponds to 7.4% and 3.5% of the original and new segmentation respectively. The effect of the proposed segmentation is evaluated in Section 3.2.

[3] In this section, we use a subset of our complete corpus described in Appendix A.

[4] Corpus segmentation, annotation and verification were performed by Gérard Bailly, director of this PhD thesis.

3.1.2 Text Annotation

As described in Appendix A.3, the textual content of the mentioned audiobooks was provided by the Gutenberg Project[5]. The original text includes abbreviations and numbers. We pre-processed the text to spell out these cases. Most frequently used abbreviations in French are "M.": "Monsieur", "Mlle": "Mademoiselle", "n°": "numéro" and "etc": "et cetera". There are no strict pronunciation rules for years in French, so numbers were spelled out according based on the audio recording of the speaker NEB[6] ("1838" can either be pronounced "dix-huit cent trente-huit", or "mille huit cent trente-huit"). Two punctuation marks were also replaced to stand as a single unambiguous character: ellipsis "..." was replaced by "~" and quotation dash, indicating turns in dialog "−−" by "¬".

3.1.2.1 Punctuation Marks as Markers of Prosodic Patterns

We introduce the symbol "§" to annotate end of paragraphs. This punctuation mark is introduced after the last punctuation mark preceding each carriage return. Ends of paragraphs are accompanied by phrasing patterns of NEB, that are worth highlighting in the training corpus. For instance, Table 3.2 shows F0 and elongation [Barbosa & Bailly, 1994] of the final syllable before ends of paragraph vs. paragraph-internal periods, as well as their values for the following syllable. The last syllable is generally longer before the end of paragraph, and the F0 gap across the boundary is increased (6.45 vs. 5.11 semitones respectively). The introduction of the symbol "§" therefore provides a way for the model to learn this modulation of duration and pitch at the end of the sentence by relying on a specific character in the input sequence.

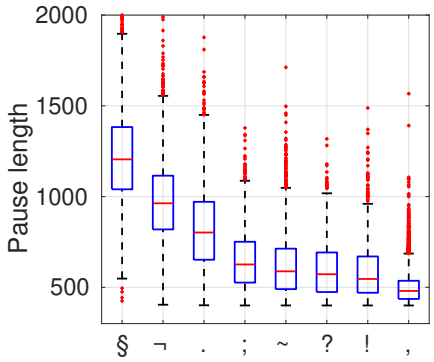
Table 3.2: Comparison of ΔF0 and elongation of syllable around ends of paragraph (.§) and intermediate periods (.). Source: [Lenglet et al., 2021]

		Syllable	
		Previous	**Following**
Elongation (%)	**.**	+184	+21
	.§	+218	+24
ΔF0 (semitone)	**.**	1.96	7.01
	.§	0.96	7.41

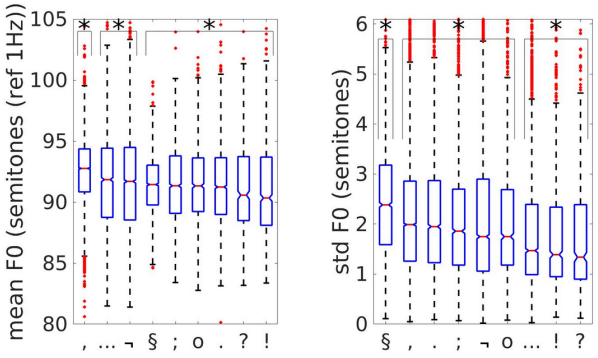
Similarly, these specific prosodic patterns associated with punctuation marks extend further than the first syllable of the next utterance. Punctuation marks help the reader to anticipate pauses and emphasis. Careful readers rely on these cues to anticipate their breathing pattern and follow-up intonation [Winkworth et al., 1994]. Fig. 3.2a indicates the distribution of pauses duration between utterances associated with the 8 most frequent punctuation marks in the corpus. The pauses made by the speaker were relatively consistent with each punctuation mark. Note that the duration of pauses between utterances is computed on the new segmentation of the corpus discussed in Section 3.1.1, so the minimum duration of pauses between utterances is 400 ms.

[5] https://www.gutenberg.org/

[6] All transcriptions and alignments were hand-checked by Gérard Bailly, supervisor of this PhD.



(a) Pause duration distribution by preceding punctuation mark. Durations are given in ms.



(b) Mean F0 and standard deviation on the first third of utterance after each punctuation mark. o stands for all other punctuation marks. * denotes groups between which differences are significant.

Figure 3.2: Prosodic Patterns Around Punctuation Marks in the ground truth.

Punctuation is not the only cue on which readers plan their speech: syntactic structure and semantic content also play an important role on speech planning [Bailly & Gouvernayre, 2012]. However, these linguistic cues are based on higher levels of language understanding, which we hypothesize that they are out of the scope of the language modeling performed by the text encoder of neural TTS. Large Language Models such as BERT [Devlin et al., 2018] have been shown to capture this layer of understanding [Oplustil-Gallegos et al., 2021; Stephenson et al., 2022], but they have not been studied in the presented work.

In order to evaluate the anticipatory effect of punctuation marks on intonation, we automatically measured F0 on the beginning of utterances following each punctuation marks in our corpus described in Section 3.1.1. F0 is measured by frame on the first third of each utterance, using the software Praat [Boersma, 2001]. Some utterances begin with a punctuation mark: '«' for reporting start of dialogues, '¬' for reporting dialog turns, '(' or '[' for side notes and '"' for quotes. In this case, this initial punctuation mark is taken as reference, instead of the ending punctuation mark of the previous utterance. Mean and standard deviation results are given in Fig. 3.2b for utterances following the 8 most frequent punctuation marks. Fig. 3.2b shows 3 groups of punctuation marks that are associated with a significantly different mean fundamental frequency at the beginning of the following utterance[7]: 1) the intra-utterance short break ',' 2) suspended speech markers '...' and '¬' and 3) end of utterance '§', ';', '.', '?', '!'.

The distribution of standard deviations of pitch following each punctuation mark also indicates various ranges of expressiveness. End of paragraphs (§) is generally associated with a longer pause, followed by a rebound of pitch variations to indicate to the listener that they have to pay attention to the potentially new theme which is introduced in the next paragraph. On the other hand, commas are associated with shorter intra-utterance breaks, followed by high mean pitch as an indicator of thematic continuity. On the contrary, '?' and '!' are

[7]Pair-wise statistical differences were evaluated by Wilcoxon rank-sum tests.

mostly found at the end of simulated turns, and followed by parenthetical elements usually uttered with lower variations of pitch.

3.1.2.2 Augmenting Text Input With Preceding Punctuation Marks

The objective evaluation of prosodic patterns performed in Section 3.1.2.1 highlighted consistent modulations of pitch and duration around punctuation marks. We hypothesize that punctuation marks can therefore be seen as contextualization symbols to specifically train the TTS model on these prosodic patterns, not only as an anticipatory effect at the end of utterances, but also as an indicator of previous context at the beginning of utterances.

Thus, we augmented each utterance of the corpus with the final punctuation mark of the preceding utterance, given as initial symbol in the input sequence. As stated in Section 3.1.2.1, some utterances already start with a punctuation mark. In this case, no additional punctuation is added to these utterances. Finally, in the case of the combination of several punctuation marks at the end of an utterance (".§" for example), only the last one is introduced as context in the following utterance. This initial punctuation mark is referred as a **linking punctuation mark** in the following. Fig 3.3 illustrates the distribution of all punctuation marks seen in the corpus. Three positions are highlighted: the first character (linking punctuation marks), the ending character and intra-utterance punctuation. The effect of the proposed linking punctuation mark is evaluated in Section 3.3.

Note that the duration between utterances reported in Fig. 3.2a cannot be learned by the model with this setup. In the new segmentation proposed in Section 3.1.1, all utterances are cut with 130 ms of initial and final recorded silence. As a result, the only duration accessible by the model during training is this 130 ms of silence. However, the inclusion of this additional initial symbol in the sequence provides the model with an input character to be associated with this initial silence. This may favor the monotonicity of the attention when computing the alignment between the text sequence and the audio output for Attention-based TTS like Tacotron2 [Shen et al., 2018].

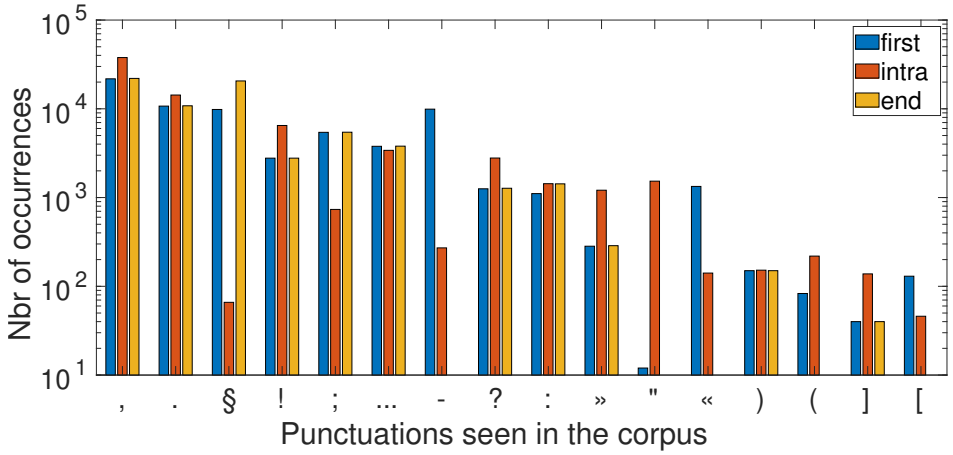


Figure 3.3: Counts of first, intra and ending punctuation marks in the new segmentation of the NEB corpus.

This annotation work can be compared to the work of Sini et al. [2018] which also augmented more than 87 h of the corpus uttered by the same speaker from LibriVox for the corpus SynPaFlex. The authors adopted a manual approach in order to provide an expert analysis of expressiveness layers: they labelled impersonated characters and their vocal personalities, as well as paralinguistic prosodic patterns. The authors distinguished two types of prosodic labels: 1) prototypical pitch contours systematically used by the speaker in a specific context ("nuance", "suspense", etc.), and 2) the set of "Basic Emotions" described by Ekman et al. [1999] and their intensity level. Our proposed text augmentation through linking punctuation marks instead relies on objective textual cues in order to convey contextual linguistic prosodic patterns between utterances. The paralinguistic annotations are not in the scope of the study described in this chapter.

3.2 Evaluation of the impact of the segmentation

Because of time constraints at the moment of this experiment, only Tacotron2 is evaluated in this chapter. The results presented in this section are extracted from Lenglet et al. [2021]. Only the main results of this paper are discussed in this section. For further information, this paper is included at the end of this manuscript.

3.2.1 Experimental Setup

Six Tacotron2 models are trained for this experiment:

- O **and** O_g are trained on the original segmentation from M-AILABS [Solak, 2019] for 200 epochs.
- N **and** N_g are trained on the new segmentation proposed in Section 3.1.1, with orthographic inputs only, for 200 epochs.
- P **and** P_g are trained on the new segmentation proposed in Section 3.1.1, with both orthographic and phonetic inputs for 100 epochs, since each epoch corresponds to twice the number of utterances of the orthographic models.

Models annotated $_g$ are fine-tuned with the Gate Loss Correction (GLC in short, see Section 2.1.2 for further details), ensuring that the EoS is properly triggered. Before the last quarter of epochs, only one model is trained. During the final quarter, models annotated $_g$ follow the GLC fine-tuning procedure, whereas others follow the standard procedure.

The number of epochs is modified from the training procedure described in Appendix B. To compare models at equivalent training time, the models without phonetic inputs are trained for twice as many epochs as the models with mixed-representations. Also, all models fine-tuned[8] from the English model trained on LJSpeech shared by NVIDIA[9]. Because of memory

[8]We initially thought that fine-tuning from an English pre-trained model could help to produce better quality syntheses. The number of epochs was actually sufficient to train the models from scratch. Models were then trained from scratch in later experiments.

[9]https://drive.google.com/file/d/1c5ZTuT7J08wLUoVZ2KkUs_VdZuJ86ZqA/view

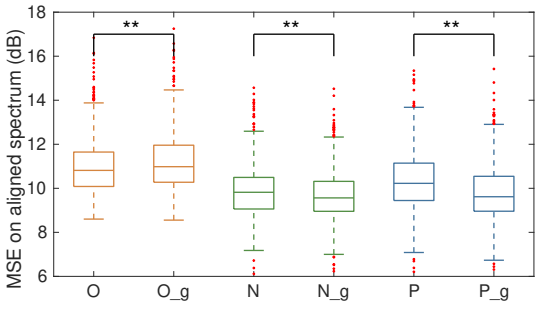
issues with longer utterances in the original M-AILABS segmentation, the batch size is set to 32 for all models. The vocoder used is WaveRNN, described in Appendix D.1.

5% of the corpus was randomly selected as the test set (903 utterances), taken from the common set of utterances between the original M-AILABS segmentation and the new proposed segmentation.

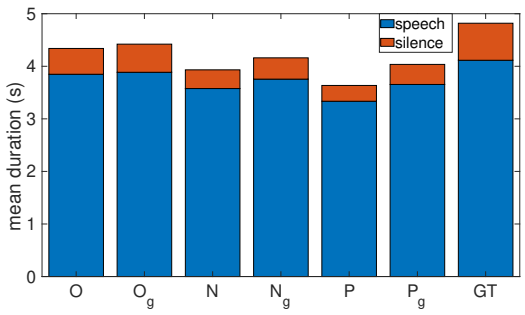
3.2.2 A Trade-Off between Spectral Accuracy and Phrasing

The main objective measurements are reported in Fig. 3.4. This evaluation reveals two opposite effects of the new proposed segmentation:

- First, Fig. 3.4a shows that the segmentation of the corpus into shorter utterances improves the spectral accuracy[10]. A one-way ANOVA confirmed the statistical effect of the model on the computed MSE error ($F = 246.5$, $p < 0.001$). Tukey-Kramer multiple comparisons show that all pairs are statistically different, except P_g/N and P_g/N_g. The gate loss correction has a significant impact on all models.
- On the other hand, Fig. 3.4b illustrates that the models trained on the new segmentation produces shorter syntheses compared to the original segmentation. This effect is mitigated by the addition of the GLC. This fine-tuning process tends to elongate synthesized utterances. Silences are elongated more than speech portions, which results in an overall lower speaker rate.



(a) Mean squared error between models and ground truth, calculated on mel-spectrograms aligned with dynamic time warping. ** indicates a significant effect of the gate loss correction according to Tukey-Kramer test ($p < 0.05$).



(b) Mean utterance duration averaged over the test set for each model. Ground truth values are showed for comparison.

Figure 3.4: Objective Evaluation of the Segmentation Effect.

[10]MSE between the synthesis and the Ground-Truth, computed on mel-scale spectrograms aligned with DTW [Kubichek, 1993].

Note that the speaking rate of all models is significantly higher than the ground truth (GT). Interestingly, the general elongation produced by the addition of the GLC also benefits the spectral accuracy on the new segmentation, but produces higher errors on the original segmentation.

Longer pauses observed with O and O_g probably result from intra-utterance pause frequency and duration in the original segmentation provided by M-AILABS. In that case, models are trained on audio clips that sometimes contain pauses longer than 1 s, and thus reproduce that behavior during inference. On the contrary, the re-segmentation processing avoids intra-utterance silences longer than 400 ms, resulting in an uninterrupted synthesis.

3.2.3 Multi-dimensional Evaluation of Perceived Differences

Following the results of the objective evaluation, only the three models with the GLC (O_g, N_g and P_g) were evaluated during perceptual tests. A MUSHRA-like test [International Telecommunications Union, 2003] was conducted online on 44 participants recruited via Prolific [Palan & Schitter, 2018]. Participants were asked to rate the overall quality on a scale from 0 (very bad) to 100 (excellent). No explicit reference was given to the participants, but the vocoded Ground-Truth was included as hidden high anchor. Results of this perceptual experiment are reported in Fig 3.5. This experiment showed a small but significant preference by participants for O_g compared to N_g, which could indicate that training Tacotron2 on the proposed segmentation does not produce the expected benefits.

As discussed in Section 1.3, MOS and MUSHRA provide limited information on the actual differences between the evaluated systems: asking participants to rate the synthesis quality without specific definition of what was expected was probably a mistake. However, in order to explore the underlying factors of the performed evaluation, we further used the recorded

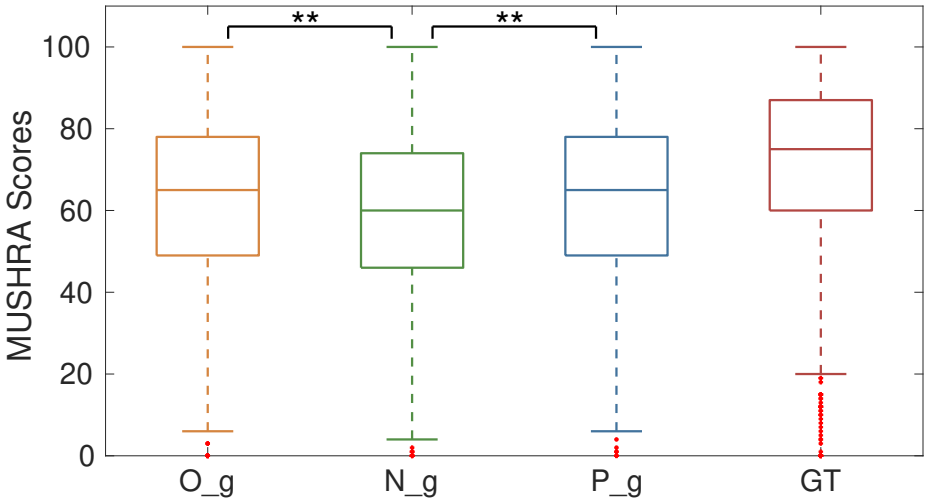


Figure 3.5: MUSHRA results. ** indicates a significant difference between models ($p < 0.05$).

MUSHRA scores to perform a multi-dimensional analysis of distances computed between each model and *GT*. These distances are evaluated against both objective and subjective measurements:

- **Subjective distances:** absolute score differences between all possible condition pairs evaluated in the MUSHRA, averaged across all participants and all utterances.
- **Objective distances:** MSE between all possible conditions pairs computed on mel-spectrograms aligned by DTW [Kubichek, 1993]. Objective distances are averaged across all 903 utterances of the test corpus.

These two distances matrices are then projected in two independent 2-dimensional spaces using classical Multi-Dimensional Scaling (MDS) [Kruskal & Wish, 1978]. To give a better idea of the impact of the GLC, both corrected and non-corrected models were also included in the objective MDS. Subjective and objective MDS (respectively named MDS_S and MDS_O in the following) are given in Fig. 3.6. As a tool to interpret the dimensions exhibited by the two MDS, correlations between acoustic measurements and the components of both MDS are estimated. Five acoustic metrics are selected:

1. The **Mean Square Error (MSE)** on aligned spectra computed in Fig 3.4a
2. The **Speaking Rate (SR)** in phones per second.
3. The **mean Pause Duration (PD)** in seconds.
4. The **mean Fundamental Frequency (mean F0 in semitones)** and the **standard deviation of F0 (std F0 in semitones)** computed by utterance and averaged across the corpus.

MSE and mean F0 evaluate the spectral accuracy of the predicted mel-spectrograms. The speaking rate and the pauses duration are related to phrasing. std F0 is evaluated as a supra-segmental expressive cue. Correlation coefficients betwen the coordinates of models on the MDSs and these acoustic features are given in Table 3.3.

Correlation coefficients indicate that prosodic cues like pause duration and standard deviation of F0 are closely related to the second component of MDS_O, but to the first component of MDS_S. On the other hand, spectral accuracy measurements MSE and mean F0 are correlated to the first component of MDS_O, and conversely to the second component of MDS_S, even if this tendency is not significant for MDS_S. In short, two main dimensions emerge in both evaluations: segmental accuracy (MSE and mean F0) and supra-segmental prosodic factors (phrasing and std F0). The axis inversion (and associated portion of variance explained) tends to show these dimensions are not given the same importance in the perceptive judgment as in the objective measurement.

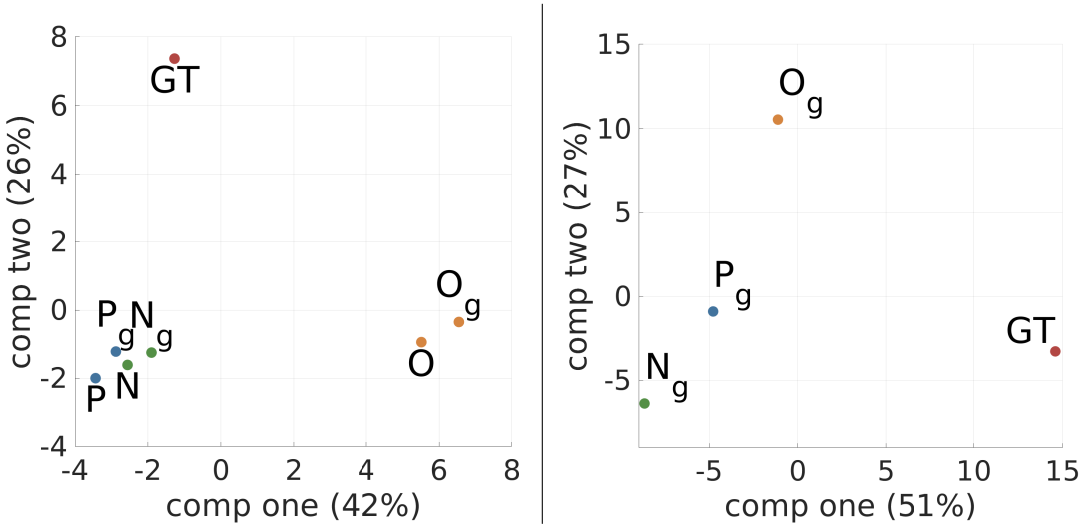


Figure 3.6: Multi-dimensional scaling of distances between pairs conditions. Left and right graphs show objective and subjective distances respectively. Proportions of variance explained are given for each component.

Table 3.3: Correlation coefficients between objective measurements and components of MDS. * and ** indicate $p < 0.1$ and $p < 0.05$ respectively.

MDS	Dim	objective measurements				
		MSE	SR	mean PD	mean F0	std F0
Obj	1	**0.90****	-0.47	0.44	**-0.71***	-0.06
	2	0.63	**-0.74***	**0.89****	-0.43	**0.97****
Subj	1	0.89	**-0.93***	**0.96****	-0.50	**0.98****
	2	0.97	-0.02	0.13	-0.83	-0.17

3.2.3.1 Notes on mixed-input representations

The ablation study presented in this section also compared the impact of training on mixed representations with training characters only. Fig. 3.5 summarizes the scores given by participants to each model. The model P_g, trained on mixed representations, performs significantly better that N_g. The multi-dimensional analysis of results presented in Section 3.2.3 indicates that P_g outperformed N_g on all aspects. Additionally, this evaluation does not take advantage of the possibility for P_g to use phonetic inputs at inference. These results are very promising for mixed-representations, and motivate us to reproduce this procedure on the experiment presented in Section 3.3, as well as all following experiments.

3.2.4 Discussion: Interesting Insights but Restricted Experimental Setup

The presented results advocate for the importance of the data segmentation on the performance measured on Tacotron2, which is often underlooked in the literature. Through the proposed multi-dimensional analysis of measured objective and subjective metrics, we have shown that the way speech data is segmented impacts both quality and expressiveness in opposite directions. The proposed segmentation in shorter utterances favors spectral accuracy, which was likely not the decisive factor in perceptual judgments.

However, this evaluation may not be representative of the potential of the shorter segmentation. First, in order to evaluate the models on a common test set, test utterances were selected as belonging to the intersection of the two segmentations. This common set is not representative of the distribution of duration in the new segmentation, with a mean duration of 4.06 s, above to the third quartile of the proposed segmentation: 3.95 s. This selection implicitly favors the original segmentation, by evaluating samples in the middle of its training distribution. Additionally, the main benefits of the shorter segmentation is not always evaluated on well-formed isolated utterances. By focusing on smaller portions of speech, the proposed segmentation does focus on the spectral quality, but these portions should then be concatenated into longer forms of speech. In that case, the phrasing has to be reconstructed at inference by managing the duration of silences between the chunks.

The role of punctuation marks in this post-hoc concatenation is crucial, which is why the punctuation is given a particular focus in Section 3.3. This concatenation process in longer forms of speech was not evaluated in this experiment. Due to its benefits on the spectral prediction, we decided to apply this segmentation process to the whole dataset in later experiments. Alternative segmentation processes may be explored in future work: in order to maximize the useful linguistic context within isolated utterances, automatic segmentation may benefit from the use of pre-trained Large Language Models. The dataset could be segmented into semantically consistent units instead of solely based on silences, ultimately reducing the amount of overly short utterances (“Ah!”, “¬Oui.”, etc.).

Nevertheless, a few interesting results came out of this study. The multi-dimensional analysis of perceptual results introduces relevant nuances to the MUSHRA results. When asked a non-specific question, participants tend to favor the expressiveness of the synthesized speech compared to the spectral proximity with the natural voice. This result was to be expected: the natural voice is not the only possible speech production for a given text. As such, the objective evaluation of the proximity of the synthetic speech with the natural voice is just one indicator of the ability of the synthetic models to produce speech-like spectral features. Differences between models may be too marginal to favor any of the studied models.

Objective evaluation has also confirmed the benefits of the Gate Loss Correction (GLC) proposed in Section 2.1.2. This fine-tuning process improved both spectral and phrasing behaviors, in particular for short utterances (improvements were only found for phrasing in the original segmentation).

Others researchers have proposed other segmentations of the corpus for the Blizzard Challenge 2023, with longer utterances and fine-tuning on concatenation of utterances to better

model long-form text [Xu et al., 2023]. Our segmentation in smaller utterances may help in reducing the overall reconstruction loss by specializing the model on a easier task, but this may not be the usual case of usage of TTS, and thus may cause unnatural behaviors at inference (quick speaking rate, too rare pauses between words, unnatural phrasing...). Note that the Blizzard Challenge did not address the synthesis of short utterances: test utterances were between 100 and 200 characters.

3.3 Evaluation of Linking Punctuation Marks

We describe below our attempt to restore linguistic prosodic cues through the contextual information provided by the linking punctuation mark. We hypothesize that the ending punctuation mark of the previous utterance should convey relevant information regarding the production of appropriate speech features for the next one. When replicating this punctuation mark at the beginning of the next utterance, this initial punctuation is seen as a label, which stands for the initial context of the utterance. The character embedding associated to this punctuation mark is expected to modify the short-range context of adjacent characters, as well as the long-range context of the whole utterance. Thus, it should bias the text-encoder output and the audio output prosody. The understanding of phrasing patterns associated with punctuation is also a requirement to generate long form speech by the concatenation of smaller portions of syntheses.

To assess the benefits of the proposed linking punctuation, we conducted an experiment on Tacotron2. Our hypothesis is threefold:

H1: adding the linking punctuation mark from the previous utterance provides relevant context information which Tacotron2 can use to improve its spectral predictive performance.

H2: using the linking punctuation mark identical to the original punctuation mark of the extract (congruent punctuation) should produce more adequate prosodic patterns than using other punctuation marks (non-congruent punctuation). Similarly, a model trained on congruent punctuation marks should produce more natural samples than a baseline model trained without linking punctuation marks, since speaking style will be averaged during the training of this baseline model.

H3: each linking punctuation mark induces a particular output prosody.

3.3.1 Experimental Setup

Two Tacotron2 models were trained for this experiment:

- ***Baseline***: Tacotron2 implementation described in Appendix B. This model is trained on both orthographic and phonetic inputs.
- ***Contextual***: The same Tacotron2 implementation, but the corpus has been augmented with the linking punctuation marks.

Both models are trained on the corpus described in Section 3.1. This subset only includes NEB, for a total amount of 32:40:53 (hh:mm:ss) of audiobooks recordings. Both orthographic and phonetic inputs are alternately used during training.

Both models follow the training procedure described in Appendix B, except that the models are trained for 200 epochs instead of 100. Similarly to the experiment described in Section 3.2, the models are initially trained using warm-start from an English checkpoint trained on LJSpeech and shared by NVIDIA[11].

We designed this experiment to evaluate how well contextual information could be conveyed by linking punctuation marks. The issue when rating utterances out of context generally comes from the multiplicity of valid intonations for a given textual content. By providing some initial context preceding the stimulus to evaluate, we hope that the listeners will be able to attune their mental representation to what the synthesis should sound like, in accordance with Latorre et al. [2014].

Thus, we selected our test set to match a specific pattern which will enable us to evaluate stimuli in context. Our pattern is the following: 1) Two successive utterances; 2) the first one starts with a full stop ('.', '¬' or '§') and the second one ends with '.§'; 3) the linking punctuation mark between these utterances is one of the 8 most popular punctuation marks, which are showed in Fig. 3.2a. These two utterances are called *'base'* and *'target'* respectively. The context is established by the base, which is kept fixed during evaluation. We excluded base-target pairs if one of them contains at least one intra-utterance punctuation mark. We identified 1362 such base-target pairs. When concatenating the base and the targets, a pause is inserted between the two chunks. The duration of this pause depends on the linking punctuation mark, and is set as the mean of the measured duration distributions reported in Fig. 3.2a. Fig. 3.7 illustrates an example of such base-target pair. We used WaveRNN as vocoder (see Appendix D.1).

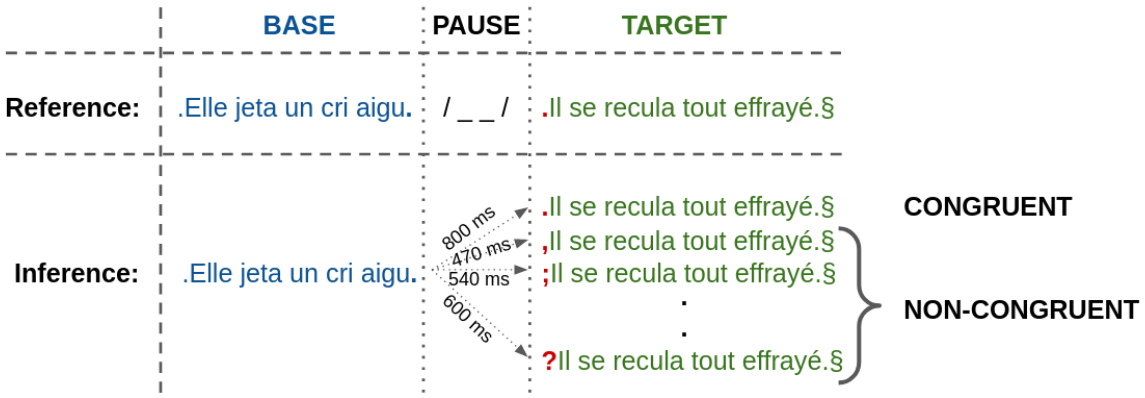


Figure 3.7: Example of a base-target pair. Their delimiting punctuation mark found in the corpus defines the congruent linking punctuation mark. At inference, any linking punctuation mark can be used; the congruent one is expected to better match the Ground-Truth target. The duration of the pause matches the mean duration measured by punctuation mark in Fig. 3.2a.

[11] https://drive.google.com/file/d/1c5ZTuT7J08wLUoVZ2KkUs_VdZuJ86ZqA/view

3.3.2 Learning Performances

To explore **H1**, we evaluated the learning performances of ***Contextual*** compared to ***Baseline***. After 200 epochs, we compute the total loss per batch of each model on the whole training corpus. The total loss is the addition of gate, spectrum and postnet loss. Batch segmentation is the same for both models. Loss evaluated on each batch is shown in Fig. 3.8.

We compared model performances with a one-way ANOVA. The ***Contextual*** exhibits a significantly lower global loss than ***Baseline*** ($F = 1200$, $p < 0.001$). These results support **H1**: the proposed model effectively uses the contextual information provided by the linking punctuation mark to bias the synthesis towards ground-truth. It is likely that the learned character embedding of the linking punctuation marks benefits the short- and/or long-term dependencies computed by the text encoder, providing better hidden acoustic features to be decoded as mel-spectrograms. Note also that this extra initial character may also help the attention mechanism, by providing an input symbol to match the initial 130ms of silence in the recordings.

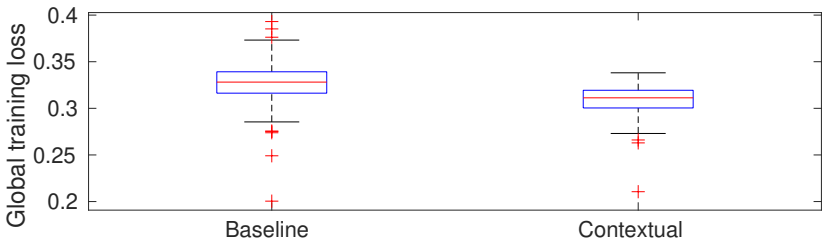


Figure 3.8: Global loss per batch on the training corpus after 200 epochs.

3.3.3 Subjective evaluation in Context (congruent vs. non-congruent)

We evaluate **H2** with a perceptual test performed online using the webMUSHRA framework [Schoeffler et al., 2018]. We selected 48 base-target pairs for the listening test (6 examples for each punctuation mark mentioned in Table 3.2a), after exclusion of mispronunciations and base-target pairs longer than 20 words. Every base was synthesized using ***Contextual*** in prediction (teacher-forcing), to ensure the best possible synthesis with our model. Each target was then synthesized under 10 conditions: 1) a baseline target with ***Baseline***, 2) a hidden reference with congruent punctuation using ***Contextual*** in prediction (teacher-forcing), 3) 8 linking punctuation targets, with the 8 punctuation marks given in Table 3.2a, i.e., the congruent and 7 non-congruent ones. The linking punctuation targets are generated with ***Contextual*** in inference mode.

Participants were separated in 2 groups, each group listened to 24 out of the 48 base-target pairs. For each pair, participants were given the original text input, and were asked to evaluate the 10 given conditions in a MUSHRA setup [International Telecommunications Union, 2003] according to the adequacy between each condition and their expected pronunciation of the given text. No explicit reference was given during the listening, so that participants were only

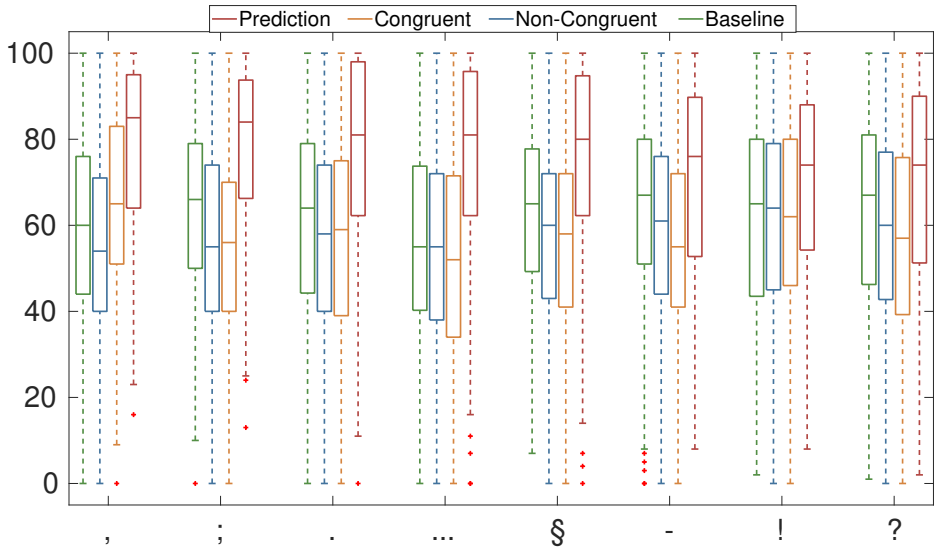


Figure 3.9: MUSHRA results by base punctuation mark. Results of non-congruent punctuation marks are averaged.

asked to judge the contextual accuracy of the target. The experiment began with 5 minutes of training during which participants listened to a variety of synthesis that they heard during the experiment and learned how to use the webMUSHRA interface. 61 participants recruited on Prolific [Palan & Schitter, 2018] and aged 18-59 took part in the experiment. They were French native speakers, and had little or no previous experience with listening tests. Results of the MUSHRA are given in Fig. 3.9.

We compared the scores of each condition for each base punctuation with pair-wise Wilcoxon rank sum test. The prediction is ranked significantly higher than the other three conditions. ***Baseline*** is ranked significantly higher than the congruent and the non-congruent conditions for ';', '§', '¬' and '?'. For the full stop '.', only non-congruent punctuation marks are significantly lower than ***Baseline***. For the comma ',', the congruent punctuation mark is ranked significantly higher than ***Baseline*** and non-congruent punctuation marks. All other differences are not significant.

Overall, only the comma supports **H2**. Several hypotheses can explain this lack of improvements of our **contextual Tacotron2** over the **baseline** for other linking punctuation marks. The base-target pattern we have chosen may not be representative of the training corpus, which made it harder for our model to generate appropriate synthesis. Plus, all linking punctuation marks do not have the same number of occurrences in the training database. In particular, the comma is much more common than the other punctuation marks, as shown in Fig. 3.11a. In addition, some of the linking punctuation marks are used in various contexts: for example '¬' is used prior to the speaking turn of a character. Characters are portrayed by NEB in a very expressive manner, that is different from one character to another. As a result, averaging speaking style according to '¬' may lead to unnatural speech or unexpected style.

3.3.4 Linking Punctuation Effects

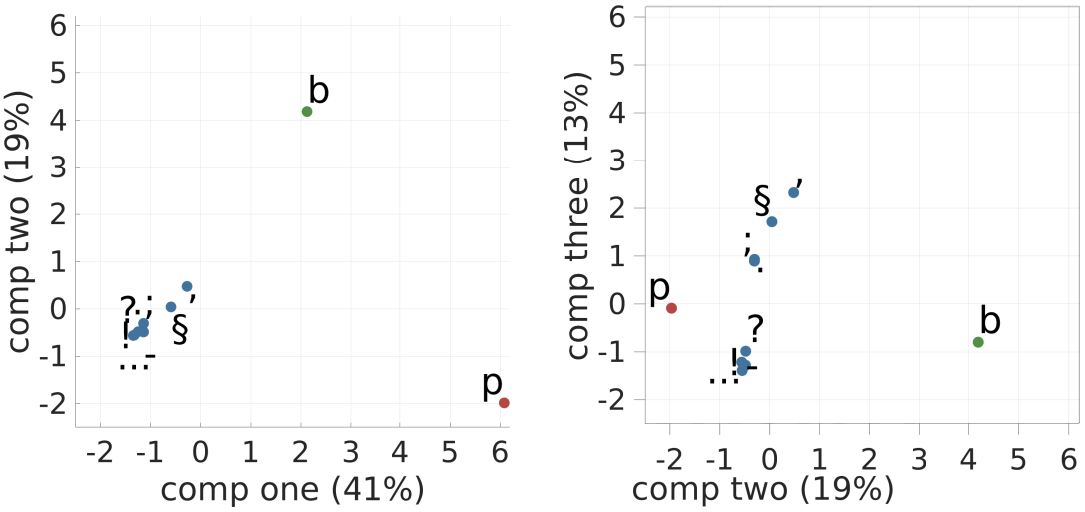
To study **H3**, we computed distances between the 10 variations of targets previously mentioned, taken as pairs. This amounts to 45 distinct pairs per target, computed as an upper-half distance matrix. This calculation evaluates the ability of ***Contextual*** to generate a variety of speech according to the linking punctuation mark used. Remember that they impact the utterance **following** them. Two distances matrices were evaluated through subjective and objective measurements:

- **Subjective distances:** absolute score differences between all possible condition pairs evaluated in the MUSHRA from subsection 3.3.3. Scores are averaged across all participants and all targets.
- **Objective distances:** mean squared error between all possible condition pairs computed on mel-spectrograms aligned by dynamic time warping (DTW) [Kubichek, 1993]. Objective distances are averaged across all 681 targets of the test corpus.

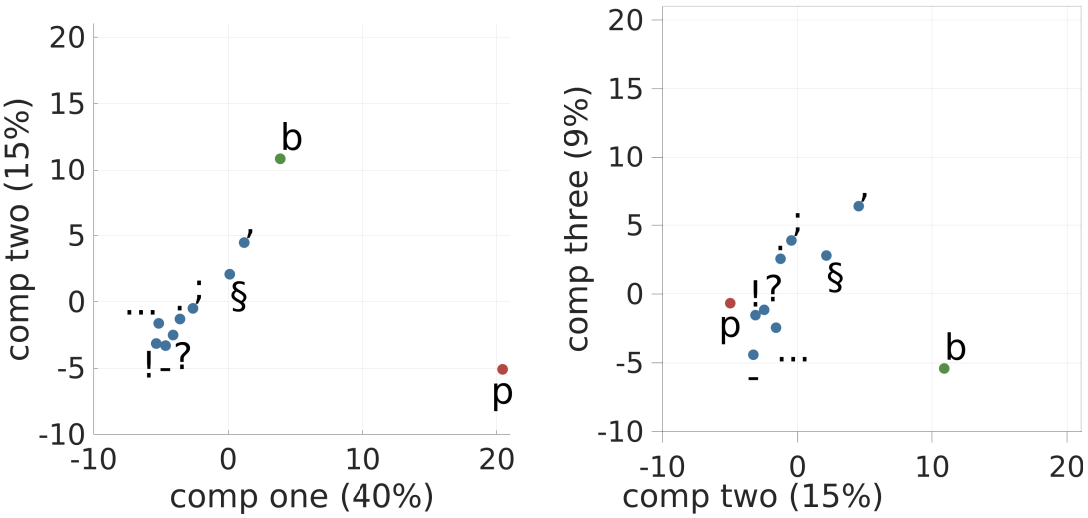
Then, we projected the two obtained distances matrices in a 3-dimensions space using classical Multi-dimensional scaling (MDS) [Cox & Cox, 2008]. The wider the dispersion of linking punctuation marks in the MDS, the more our contextual Tacotron2 is able to generate variations in speech. Subjective and objective MDS are given in Fig. 3.10.

The subjective and objective MDS show very similar patterns. That means that the MSE computed on synthesis is a strong prior indicator of the subjective scoring. Moreover, subjective results by congruent punctuation mark in Fig. 3.9 can be used to interpret the dimensions of the MDS regarding **H3**:

- From Fig. 3.9, on average, the **baseline** scored higher than any linking punctuation of our **contextual Tacotron2**; thus it is closer to the prediction. On Fig. 3.10a and 3.10b, the hierarchy of linking punctuation marks of the first component (x-axis on the left graph) matches the average score obtained by each condition during the experiment, suggesting that the first component stands for the general quality of speech, which is what participants mostly evaluated.
- The prediction is clearly the most expressive condition as it was generated with teacher-forcing. On the contrary, ***Baseline*** was trained on the entire corpus, so speaking styles were averaged across all utterances, and no particular style selection manner was added. On the other hand, utterances synthesized with ***Contextual*** employ a wider variability of expressions. Given that Fig. 3.10 shows proximity between prediction and linking punctuation along the second component, it suggests that this component stands for the expressiveness of the synthesized speech.
- The third component spreads out the different linking punctuation marks: on the one hand expressive punctuation marks (‘¬’, ‘?’, ‘...’ and ‘!’) that are mostly used in dialogues, and on the other hand non-expressive punctuation marks (‘,’, ‘§’, ‘.’ and ‘;’) that are mostly used in narrative utterances. ***Baseline*** averages different punctuation marks, thus is neutral regarding that aspect.



(a) Objective MDS: plane 1 (left) and 2 (right)



(b) Subjective MDS: plane 1 (left) and 2 (right)

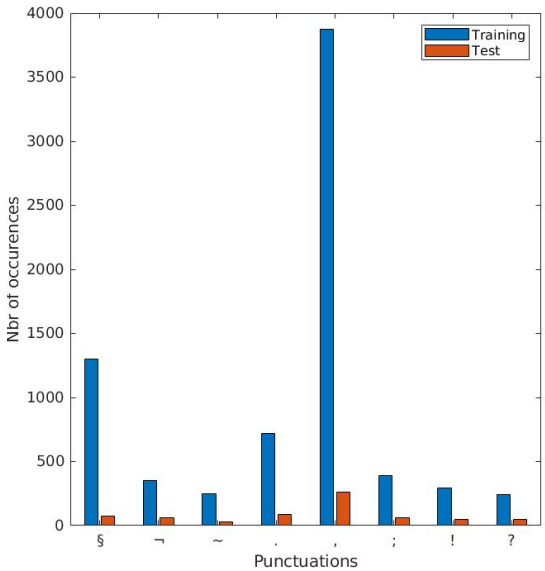
Figure 3.10: Multi-dimensional scaling of distances between paired conditions. Upper and lower graphs show objective and subjective distances respectively. Both show first and second factorial designs respectively. Proportions of variance explained are given for each component. b and p stand for baseline and prediction respectively.

3.3.5 Discussion: Generation of Uncontrolled Variability

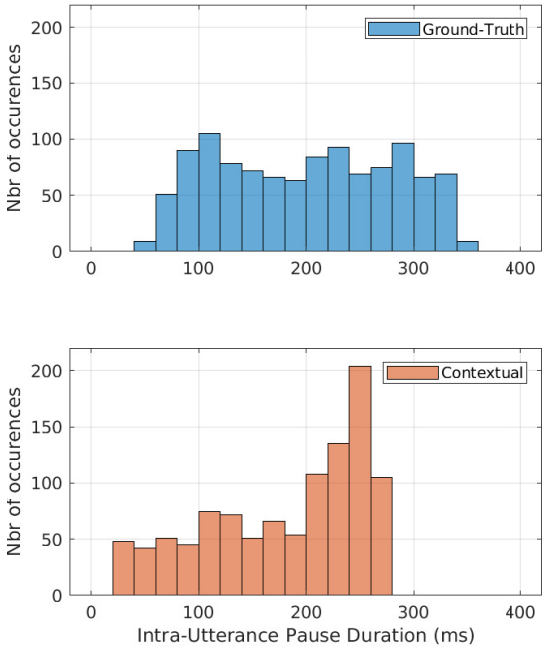
This proposed experiment showed that using the punctuation mark ending the previous utterance to bias the synthesis of the current utterance enables the Tacotron2 model to generate a wider variability of speaking styles. However, we were not able to take advantage of this variability at inference, which reduces the benefits of the proposed augmentation method.

Improvements are still to be found to use this variability to actually improve synthesis in context. Training on corpora that are more balanced in punctuation marks could increase synthesis accuracy in these cases, as seen with the comma. Nevertheless, increased variability observed with the proposed contextual Tacotron2 opens new perspectives on how to use a single character at the beginning of utterances to bias the whole utterance prosody. Other characters that do not necessarily appear in the original text may also be used for this purpose, such as labels or emoticons. Characters and voice identity labels from SynPaFlex [Sini et al., 2018] are interesting candidates to test this method.

Even if they were excluded from the test set, intra-utterance punctuation marks are very common in the corpus. A finer analysis of the duration associated with these intra-utterance pauses in the Ground Truth (see Fig 3.11b) reveals a tri-modal distribution with harmonic log-scaled modes: 1) ∼120 ms, 2) ∼240 ms and 3) ∼480 ms found by Bailly and Gouvernayre [2012] and Campione and Véronis [2002]. This behavior is not reproduced by our model, which favors pauses of the intermediate length of ∼240 ms. Instead of relying solely on punctuation to annotate these pauses, we could have used additional symbols for each type of pause seen in the corpus in order to help the model to learn this behavior.



(a) Number of initial occurrences of the 8 most common punctuation marks in training and test corpus.



(b) Duration distribution of pauses in the corpus VS at inference for ***Contextual***. In the 1362 utterances of the test set, we counted 1095 and 1056 intra-utterance pauses for the Ground-Truth and ***Contextual*** respectively.

Figure 3.11: Comparison of the training and test sets.

Despite the limited results of the proposed linking punctuation procedure, this text-augmentation was applied to the entire corpus described in Appendix A and reproduced in all following experiments. From a training perspective, this linking punctuation mark improves the reconstruction loss of Tacotron2. It favors the monotonicity of the attention mechanism, which helps interpreting the attention maps of Tacotron2 as described in Section 2.2. This initial symbol is also a requirement to train the duration predictor of FastSpeech2 on this initial 130 ms of silence. We have tried to introduce two additional symbols (out of the existing set) to annotate these initial and final silences ("0" and "1" respectively). However, punctuation marks provided better reconstruction losses and more consistent attention maps, so we stuck to the proposed process.

3.4 Discussion: Limited Expressive Control Through Text Sequences

The proposed methods mainly focus on the structuring of the data and how to present them to the TTS model in order to improve the modeling of linguistic prosody. The segmentation of the training dataset into utterances, massively under-explored in the literature, was proved to impact both the spectral reconstruction capabilities and the expressiveness of the TTS model. We ultimately opted for a segmentation into shorter utterances, which favors the spectral accuracy at the cost of more natural phrasing. We believe that phrasing may instead be predicted based on contextual intra-utterances cues like the punctuation. We have demonstrated that the punctuation highlights prosodic patterns that can be learned by neural TTS as a contextualization tool. However, we found little success when controlling the synthesis through the proposed linking punctuation procedure.

We emphasize that the segmentation does not provide any explicit control mechanism at inference, but was expected to contribute to the enhancement of performances through the introduction of inter-utterance contextual information. The addition of linking punctuation on the other hand was expected to propose an alternative to the one-to-many mapping problem of voice synthesis. We found that the use of different punctuation marks as the initial character of the text sequence did indeed produce various prosodic patterns, but we did not manage to turn this variability of production into a viable prosodic control mechanism. We may have overestimated the contextualization capacities of the neural layers in our model.

This made us question our understanding of the learning process of neural TTS. Despite the exhibition of prosodic patterns in the Ground Truth recordings, these patterns were not reproduced by our model during inference. In order to interpret this mismatch, we need to dive into TTS model internal representations and figure out how the acoustic and prosodic information is encoded. We believe that a closer look at the features encoded into latent embeddings could help us to design more careful architectures based on a finer understanding of the unsupervised processing pipeline performed by such models. Additionally, the probing of acoustic features may provide new opportunities to bias the models toward the desired expressive control, with respect to the models' representations. We propose analytical methods to uncover TTS model embeddings in the following chapter.

Chapter 4

A Closer Look at Internal Representations of Neural TTS

Contents

Chapter Highlights

This chapter presents our analytic methods to unveil the acoustic and phonological features encoded into latent representations of neural TTS. We show how **linear probing** reveals the different dynamics of computation of acoustic features in the successive layers of Tacotron2 and FastSpeech2. Notably, this analysis highlights that mean **segmental features** like formants are encoded into the early layers of both models, which confirms the ability of the text encoder to predict the Letter-to-Sound alignment. On the other hand, **supra-segmental features** (like F0 and duration) are sensitive to the sub-tasks implemented by the models. This analysis advocates for a more careful design of sub-tasks with respect to the actual processes learned by neural layers in an unsupervised manner. We emphasize that the proposed analytic methods are not model- or feature-dependent, and thus could be universally used to better understand the underlying processes performed by neural TTS.

Related contributions: [Lenglet et al., 2022b, 2023a]

In this fourth chapter, the focus is placed on the internal representations of neural TTS, referred to as **embeddings**. Deep learning architectures, by definition, compute intermediate internal representations which encode several levels of information. Because these internal spaces are mostly unconstrained, understanding these latent representations can be challenging. The lack of interpretability of these neural models constitutes a barrier to the development of these systems in interactive environments [Gunning, 2017].

In an attempt to unveil the hidden processes implemented in these black boxes, methods for analyzing the internal representations of neural models are emerging [Burkart & Huber, 2021]. Explainable Artificial Intelligence (XAI) may seem less of an issue for synthetic speech related tasks than it is for health care. Predicting mel-spectrogram features surely does not raise the same ethical concerns than medical diagnosis. However, the benefit of XAI lies here in the potential benefits for scientific understanding of complex phenomena. Statistical learning performed by these neural models constitutes a valuable source of information about language if analyzed with the right tools and methodology. Correlations learned by deep learning models to maximize their predictive capabilities may help us to better understand human speech production mechanisms from low-level phonetic co-variations to high-level phonological organization of sounds.

Moreover, uncovering the specific tasks performed by each layer of a deep neural network also helps in designing more careful neural architectures [Zeiler & Fergus, 2014]. Neural model architectures often rely on heuristics to determine the optimal number of dimensions of latent spaces or the number of similar layers to stack to achieve one specific goal. More often than not, the final decision for these hyperparameters is based on ablation studies, which give limited insights on the reasons why some architectures perform better than other. We believe that developing methods to probe these latent spaces could help us understand how variations of architecture impact the encoding of features of interest.

Uncovering features encoded in internal neural layers is specifically interesting for TTS models, for which we observe the use of external latent representations of the input text or the output signals to achieve increasingly precise control. As seen in Section 1.2, the latest TTS models may now integrate pre-trained representations from speaker-verification tasks [Jia et al., 2018], Large-Language-Models [M. Kim et al., 2021; Shin et al., 2022] or self-supervised audio representations [L.-W. Chen & Rudnicky, 2022]. Although these representations are believed to convey some acoustic, syntactic or semantic information which could help the synthesis process, a deeper analysis of these representations should precede their integration to ensure that 1) they encode the specific information the TTS model could use to improve its prediction, but also that 2) these external representations are integrated in the appropriate layer·s in which the TTS will make the best use of this additional information.

This chapter presents the set of methods proposed to analyze the internal spaces of neural TTS models, in order to track acoustic and prosodic representations in embeddings. The linear probing procedure of acoustic representations in internal embeddings in presented in Section 4.1. In an attempt to show its universality, this procedure is illustrated on both Tacotron2 and FastSpeech2 models, for a selection of acoustic features. Results are presented for segmental and supra-segmental features in Section 4.2.

4.1 Feature Tracking in Latent Representations

The interpretation of TTS as recurrent auto-encoder architectures trained to compute self-supervised speech representations, which are biased by the text during decoding, is the foundation of the Reference Encoder widely adopted by the TTS field [Skerry-Ryan et al., 2018]. This interpretation supposes that the text encoder modulates acoustic representations set by the Reference Encoder as a function of phonetic content. As stated in Section 1.2.2, various utterance-wise contributions might be combined in order to account for multiple levels of expressiveness. However, little is known on how these representations are structured and what they encode.

With regard to utterance-wise embeddings, identity and gender have successfully been shown to be encoded by speaker representations trained for speaker verification [S. Wang et al., 2017], later used as speaker embeddings in TTS [Jia et al., 2018]. Similarly, reference embeddings were shown to linearly encode eGeMAPS acoustic features [Tits et al., 2021]. These findings advocate for the hypothesis of mean acoustic features being encoded into utterance-wise embeddings, which are modulated in time by the output of the text encoder, in order to generate the appropriate audio output.

Regarding character embeddings, we showed in Section 2.2 that internal representations of orthographic input encode the sequence of phones to produce, but that study only provided qualitative interpretations of how Tacotron2 models the duration encoded in each input symbol. In this section, we propose to extend the analysis of acoustic correlations in latent spaces proposed by Tits et al. [2019] and apply it at the text level. This analysis should provide the first insights into the local encoding of acoustic modulations performed by neural TTS.

4.1.1 Choice of Acoustic Features

Acoustic features of interest were chosen to match the proposed local scale. Despite the eGeMAPS features [Eyben et al., 2015] having shown great potential in affective computing, most parameters are measured on long time scales, generally at the utterance-level. This prevents the use of these features at the phone level. Instead, we considered a smaller set of local features which define both the spectral identity of phonemes (called **segmental features**) and wider-range prosodic variations independent of the phoneme identity (called **supra-segmental features**). The full list of features measured is given in Table 4.1. As discussed before, the text encoder is suspected to encode phonetic information, which means that formants are also likely to be encoded in its embeddings. The three prosodic features – phone duration (D), its average fundamental frequency (F0) and energy (E) – were chosen for their prevalence in expressive control in the literature [Mohan et al., 2021; Raitio et al., 2020; Ren et al., 2021; Y.-J. Zhang et al., 2019]. Additionally, since FastSpeech2 implements an explicit predictor from the output of the text encoder for each of these features, they are likely to be found encoded in latent representations. Spectral tilt, center of gravity and spectral balance were also included for their role in phonetic discrimination and voice quality.

Table 4.1: Acoustic Features tracked in latent representations of neural TTS. Semitones (st) are computed with reference 1Hz. All acoustic parameters are averaged on the sustained part of the vowels.

Type of Feature	Acoustic Feature	Abbreviation	Unit
Segmental	First Formant center frequency	F1	st
	Second Formant center frequency	F2	st
	Third Formant center frequency	F3	st
	Spectral Center of Gravity	CoG	st
	Spectral Balance around 1kHz	SB1k	dB (High Freq energy ÷ Low Freq energy)
Supra-Segmental	Duration	D	log(1 + # of frames)
	Fundamental Frequency	F0	st
	Energy	E	dB
	Spectral Tilt	ST	dB/octave (Relative Value)
	Relative Position in Utterance	RP	index ÷ input sequence length

The relative position in the sequence is included as an additional supra-segmental feature to explore how the recurrent VS parallel processing performed by the TTS architectures may need to encode relative position in various ways. Since the use of self-attention layers [Vaswani et al., 2017], a sinusoidal positional encoding is added to model sequential information. Indeed, parallel data processing cannot access on its own the relative positions of the symbols in the sequence. On the other hand, recurrent networks like LSTMs inherently generate hidden representations sequentially. As a result, the encoding of this sequential nature may not be necessary into latent representations. We will however show that relative positioning is computed at key stages of the text-to-sound mapping.

All acoustic features are measured with Praat [Boersma, 2001]. Time alignments computed by the models (duration predictions for FastSpeech2, extracted from the attention map for Tacotron2 using the procedure described in Section 2.2.2), are saved alongside the syntheses. These alignments are used to perform the acoustic analysis by phone. Because the selected features include parameters that are only measured on voiced portions of speech (i.e., F0), only vowels are taken into account when training linear predictors on these features. In this case, all parameters are averaged over the central part (33-66%) of the vowels.

4.1.2 Training of Linear Predictors on Internal Text Embeddings

This section describes the methods we propose to track acoustic and phonetic features in the intermediate embeddings of TTS neural models. These methods are applied to the two baseline models described in Chapter 2. The comparative analysis of Tacotron2 and FastSpeech2 embeddings enables us to better apprehend how the very structure of the model shapes the process of encoding acoustic parameters by the successive neural layers.

Fig. 4.1 summarizes the methodology. This procedure consists of training multiple linear predictors on intermediate embeddings to investigate how well can each acoustic and phonetic feature be predicted at any stage of the encoding/decoding process. The evaluation of the goodness of fit (R^2) of these predictors can be interpreted as indicators of the presence/absence

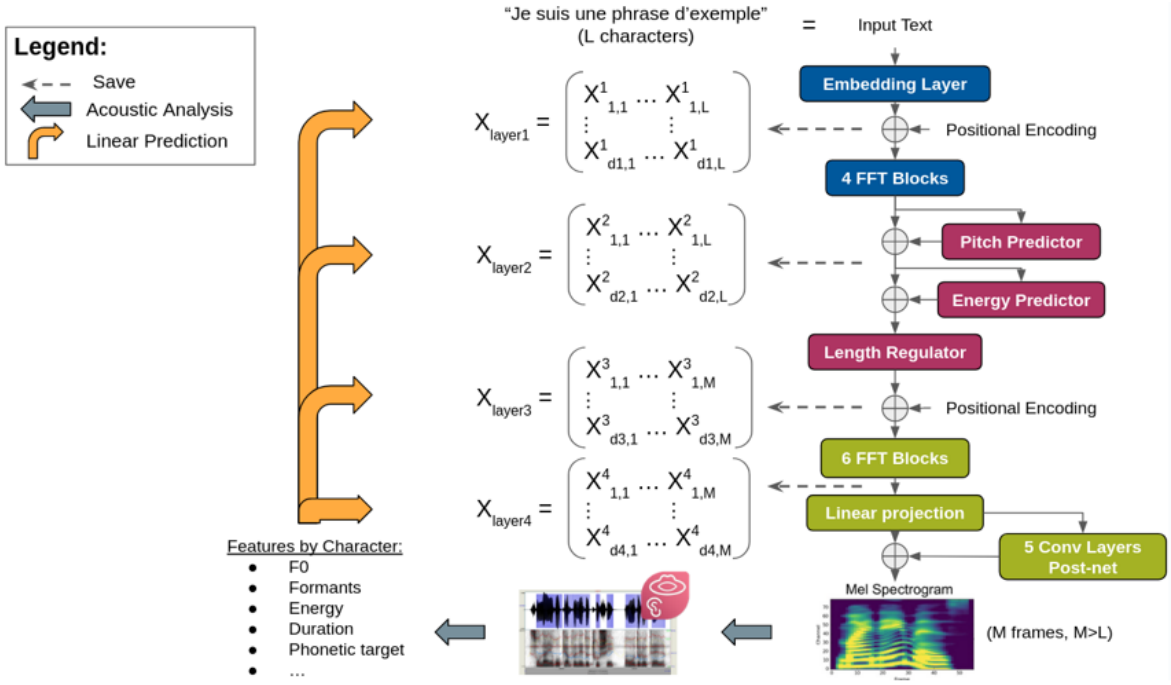


Figure 4.1: Embeddings analysis per layer: Procedure described for FastSpeech2.

of a linear encoding of each feature at any step of the prediction process. Despite the limitations of linear mapping, the proposed methodology through linear regression enables us to design linear biases (i.e., translation) to turn any intermediate latent space into an explicit control space for any predicted feature. This so-called **causal control** is explored in Chapter 5. The detailed breakdown of the proposed regression procedure is the following:

Synthesis of a subset of the training corpus: After the model is fully trained, a subset of the training corpus is synthesized in inference mode, using phone inputs. Intermediate embeddings computed in the successive layers of the encoder and decoder are saved. In order to limit the impact of outliers in audio features measured on these syntheses, only a portion of this set is used in the following procedure; we select half of the synthetic utterances whose prosodic features most closely match the corpus-averaged features. More specifically: (a) duration, fundamental frequency and energy are averaged by utterance on this synthesized set; (b) the same features are averaged on the whole synthesized corpus; (c) only half the utterances with the lowest MSE compared to the corpus means are kept for further analysis. This selection ensures that the linear predictors are trained on the most trustful part of the distribution.

Embedding Space Reduction: (model- and feature-dependent) The proposed regression can be done both in the initial latent space or on a reduced version. In most models, the intermediate latent spaces exhibit strong correlations (positive and negative) between some of their dimensions. However, multi-linear regression relies on the assumption that explanatory variables are independent, so a reduction of the latent space is here performed with Multi-Dimensional Scaling (MDS) [Kruskal & Wish, 1978]. MDS was

favored over other linear reduction techniques because we found that cosine distances between embeddings better reflects the proximity/dissimilarity between similar/different phones. In an attempt to work with accurate reduced representations while limiting the number of dimensions, only dimensions that explain at least 90% of the total variance are kept. The number of retained dimensions is thus model- and layer-dependent: models with higher dimensions of their internal spaces often need more dimensions in the reduced spaces to account for 90% of their variance.

Multi-Linear Regression by layer: Similarly to Tits et al. [2021], a multi-linear regression is computed on embeddings to 1) evaluate which features can be linearly predicted from the embeddings and 2) exhibit directions in the latent space associated with continuous variations of each feature. In contrast with Tits et al. [2021], this regression is applied on phone embeddings, instead of utterance-wise style embeddings. This regression procedure is applied for each feature, at each layer of the models. Since embeddings are up-sampled at the spectrogram frame-level in the decoder, embeddings by phone are then computed by averaging the frame embeddings sequence corresponding to each phone. This sequence is given by the automatic segmentation of the audio signal. Acoustic and prosodic features are finally estimated by least-squared multi-linear regression in the reduced spaces, using a LBFGS solver [Wright, Nocedal, et al., 1999].

Parsimonious Dimension Selection (feature dependent): The trained predictors give the lowest residual error possible in the reduced spaces, but still rely on numerous dimensions to explain each feature. To further reduce the number of dimensions involved for each feature, a parsimonious dimension selection is applied for each feature. This means that 1) The same multi-linear regression is computed by ignoring one dimension at a time. All correlations between predicted and measured acoustic features are saved. 2) The dimension that reduces the correlation the least is ignored for future regressions. 3) This procedure is repeated until the correlation reaches 1% difference with the initial best possible correlation in the reduced space. On average, this selection reduces by half the number of dimensions of the reduced space on which the regression is computed, with little impact on the predictive capabilities of the method. As a result, this procedure predicts any continuous acoustic or prosodic features by layer, following the formula 4.1.

$$\hat{P}_L = f(E_L) = E_L.A_L^P + \beta_L^P = \sum_{i=1}^{d} \alpha_{i,L}^P.e_{i,L} + \beta_L^P \tag{4.1}$$

with d the number of reduced dimensions after parsimonious selection in layer L. $\hat{P}_L$ is the approximation of any feature P at layer L. $E_L = (e_{1,L}, e_{2,L}, \ldots, e_{d,L})$ is the embedding coordinates at layer L. $A_L^P = (\alpha_{1,L}^P, \alpha_{2,L}^P, \ldots, \alpha_{d,L}^P)^T$ is the regression coefficients vector for the parameter P along the d dimensions and β_L^P the bias.

We also extended this procedure to categorical predictors. Similar to the prediction of continuous variations of acoustic features, categorical prediction indicates in which layer phonological decisions are made by the model. We trained categorical predictors to predict output phones and the presence of French liaisons from orthographic inputs, as well as the insertion of silences at word boundaries. In this case, multi-linear regressions are replaced by classification with Linear Discriminant Analysis (LDA). Output phone targets are set according to the Letter-to-Sound mapping (L2S) described in Section 2.2.3.

4.2 Linear Probing Applied to Tacotron2 and FastSpeech2

4.2.1 Experimental Setup

This study attempts to identify prosodic and acoustic features in the intermediate embeddings depending on the TTS architecture (Tacotron2 and FastSpeech2). The model architectures are fully described in Chapter 2. The training follows the procedures explained in Appendix B for Tacotron2 and Appendix C for FastSpeech2. Both architectures are trained on a single-speaker setup, using NEB data from our corpus described in Appendix A. 5% of the corpus is put aside as test set (2230 utterances). The vocoder used to compute speech from spectrograms is Waveglow (see Appendix D.2).

In order to make a fair comparison between architectures, three variants by model are trained: 1) without any prosodic predictors (except the mandatory duration predictor for FastSpeech2), 2) with prosodic predictors but without injecting the prediction through prosodic embeddings, and 3) with prosodic predictors and predicted prosodic embeddings re-injected in the decoding process. All variants are summarized in Table 4.2. Note that, contrary to FastSpeech2, the duration prediction is not used at inference time in Tacotron2 because the text-to-frame alignment is made by the attention network. Also, a version of FastSpeech2 without phonetic prediction was also trained, in order to verify the potential impact of this sub-task on features encoded in embeddings. Note that explicit predictors are expected to shape the output of the text encoder to force pitch, energy, duration and phonological information to be encoded at the output of the encoder. We hypothesized that by removing these sub-tasks, the same features should presumably still be encoded by the model, but potentially in different layers.

Name	Model	Prosodic Prediction	Prosodic Embeddings	Phonetic Prediction	Kept in Experiments
$\boldsymbol{TC}$	Tacotron2	✗	✗	✓	✓
$\boldsymbol{TC_P}$	Tacotron2	✓	✗	✓	✓
$\boldsymbol{TC_E}$	Tacotron2	✓	✓	✓	✗
$\boldsymbol{FS}$	FastSpeech2	✓	✓	✓	✓
$\boldsymbol{FS_{\setminus phon}}$	FastSpeech2	✓	✓	✗	✓
$\boldsymbol{FS_{\setminus E}}$	FastSpeech2	✓	✗	✓	✓
$\boldsymbol{FS_{\setminus P}}$	FastSpeech2	✗	✗	✓	✓

Table 4.2: Models under study. Vanilla architectures with phonetic prediction are $\boldsymbol{TC}$ for Tacotron2 and $\boldsymbol{FS}$ for FastSpeech2. $\boldsymbol{P}$, $\boldsymbol{E}$ and $\boldsymbol{phon}$ refer to the Prosodic Predictor, Prosodic Embeddings and Phonetic Predictor respectively, and $\setminus$ to the absence of this layer.

Note that Tacotron2 with re-injected embeddings of prosodic predictors ($\boldsymbol{TC_E}$) never produces satisfactory synthesis quality, because of the lack of convergence of the attention layer. We hypothesize that the recurrent LSTM layer which computes the attention weights in Tacotron2 is not fitted to handle the variability of prosodic embeddings in successive text embeddings of the input sequence. As a result, $\boldsymbol{TC_E}$ was excluded from further observations.

As explained in Section 4.1.2, after the training phase, a subset of the training corpus (5% = 2120 randomly chosen utterances) was synthesized with each model variant. All intermediate embeddings are saved, and linear predictors are trained to map acoustic features with these embeddings. The reliability of the given linear prediction is evaluated through goodness of fit for continuous features, or classification performance for classification tasks. Unless stated otherwise, all presented results are obtained using phonetic input, even if models are trained using mixed input.

The analysis of the results is organized as follows: sub-section 4.2.2 is focused on segmental features, which are key identification factors of the sequence of acoustic targets computed by the text encoder from the input sequence. Then, supra-segmental prosodic features are studied in sub-section 4.2.3, in order to better understand how prosody is modeled by the TTS architectures.

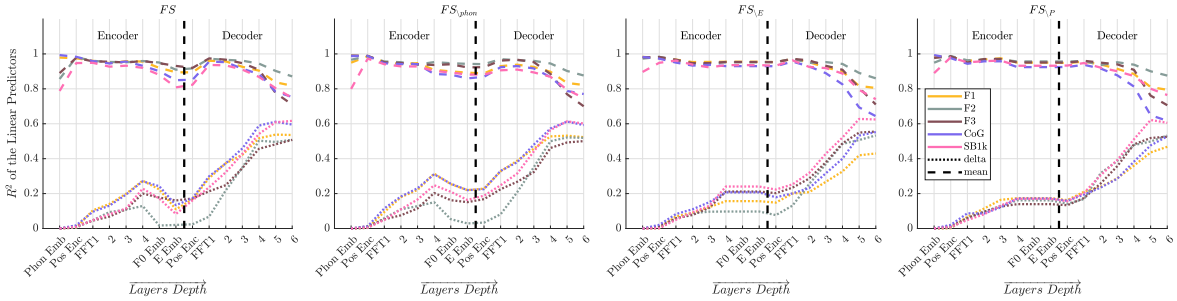
4.2.2 Segmental Features

4.2.2.1 Spectral Cues

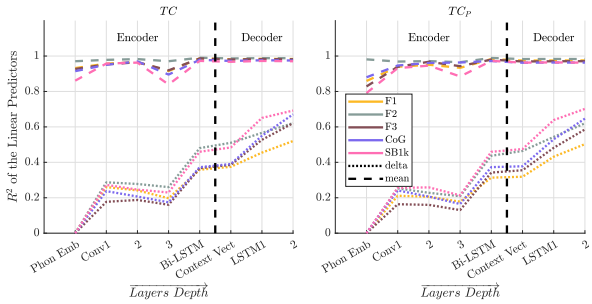
The "in-painting" of the mel-spectrogram at the output of the decoder (e.g. placement of formants and harmonics in the frequency bands) heavily relies on the segmental spectral cues being encoded into embeddings. Since spectral cues are phone-specific, before the regression, the proposed analysis considers a decomposition of spectral features in two factors: (1) the average feature of each phone class, and (2) the difference between this average feature and the actual feature computed on each phone (Δ Features). This decomposition allows us to distinguish between contributions from the chosen phone input and its contextualization through successive model layers, respectively. The goodness of fit of the multi-linear regressions by layer computed on both factors is reported in Fig. 4.2.

Fig. 4.2 indicates that the mean spectral features of each phone class is encoded right from the phone embeddings layer at the input of each model. Statistical learning performed by neural models aims at learning compact representations in order to produce predictions as close as possible to the training examples. Since there is a mapping between the phoneme-classes and the three first formants, neural models are able to encode mean formants directly into phone embeddings.

On the other hand, the differences from these mean features per phone class depend on the context (e.g. co-articulation, variations due to prosody, etc.). As a result, differences from mean features are increasingly well modeled throughout successive layers, until reaching the output of the decoder where these representations are projected into 80 dimensions to compute the predicted mel-spectrogram.



(a) R^2 by layer for FastSpeech2 variants



(b) R^2 by layer for Tacotron2 variants

Figure 4.2: Goodness of fit of the Multi-Linear regressions by layer on segmental features. The x-axis indicates the successive layers of the model (from left to right): "FFT" refers to Feed-Forward Transformer, and "Conv" to convolutional layer. Goodness of fit is expressed as the R^2 of the multi-linear regression. Acoustic features are detailed in Table 4.1. Predictions of mean spectral features are displayed with dotted lines while those of deviations from the means are displayed with dashed lines.

Interestingly, regardless of the model, the encoder alone cannot achieve contextualization. The decoder still performs a large part of the contextual encoding of spectral features, especially in the case of ***FS*** and $\boldsymbol{FS_{\setminus phon}}$, whose re-injection of pitch and energy embeddings temporally degrades spectral representations. Tacotron2 variants better encode linear representations of spectral features at the output of their encoder ($R^2 \approx 0.4$ compared to 0.2 for FastSpeech2 variants). One possible explanation of this phenomenon is that Tacotron2 autoregressive decoder performs less operations compared to the stack of multi-head self-attention blocks in FastSpeech2, which may constrain the encoder to compute representations that are closer to the predicted spectrogram. The presence of prosodic predictors does not strongly impact these observations.

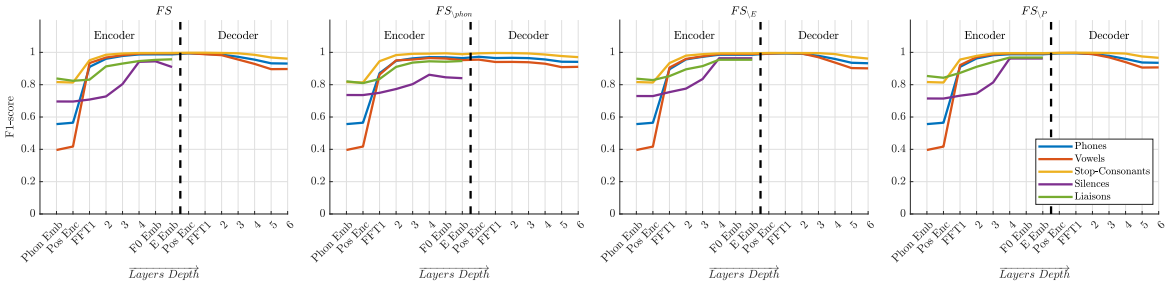
4.2.2.2 Phonetic Prediction

The early encoding of mean spectral features, which are closely linked to phone identity, in all models suggests that this early coding of phonetic representations may hold true for orthographic inputs. To explore this hypothesis, orthographic input is used in this section to produce intermediate orthographic embeddings. Linear classifiers were trained on these embeddings using Linear Discriminant Analysis (LDA). Performance of these classifiers by

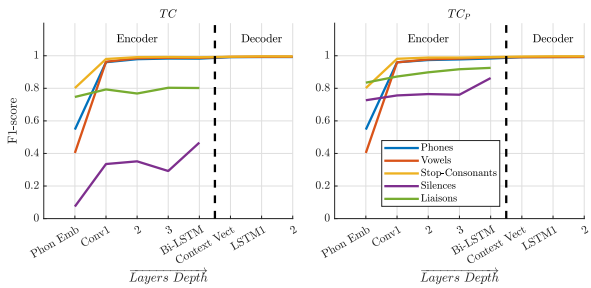
layer is reported in Fig. 4.3. This analysis complements the observations of attention maps presented in Section 2.2, which highlighted that phone duration was predicted early by the text encoder when synthesizing speech from orthographic input.

Fig. 4.3 distinguishes between classification performance on all phones (blue), vowels only (red), stop-consonants (yellow), silences (purple) and French liaisons (green). Silence and liaison predictions are limited to the encoder, since the presence of a character embedding itself in the decoder indicates that this character has a non-null duration, which mean that the silence or the liaison has been produced. Regardless of the model, phonetic representations appear in the very first layers of the encoder. Unsurprisingly, vowels need more contextualization than stop-consonants to be correctly predicted. The differences in F1-scores between vowels and stop consonants from the first embedding layer is explained by the variability of possible phonetic outputs for the corresponding characters.

Silences at word boundaries and liaisons are encoded later in the process. These two phonological behaviors rely on the encoding of the duration in the text embeddings. Indeed, the only difference between the input character "space" producing a silence or not is the duration predicted for this character (same for final consonant in a position of optional liaison). Thus, this indicates that duration only appears at the output of the encoder for FastSpeech2 when the phonetic prediction task is implemented. On the other hand, Tacotron2 shows worse performance on silence and liaison prediction. Tacotron2 relies on its attention map to predict phone duration. The attention process is autoregressive, which does not limit the



(a) F1-score by layer for FastSpeech2 variants



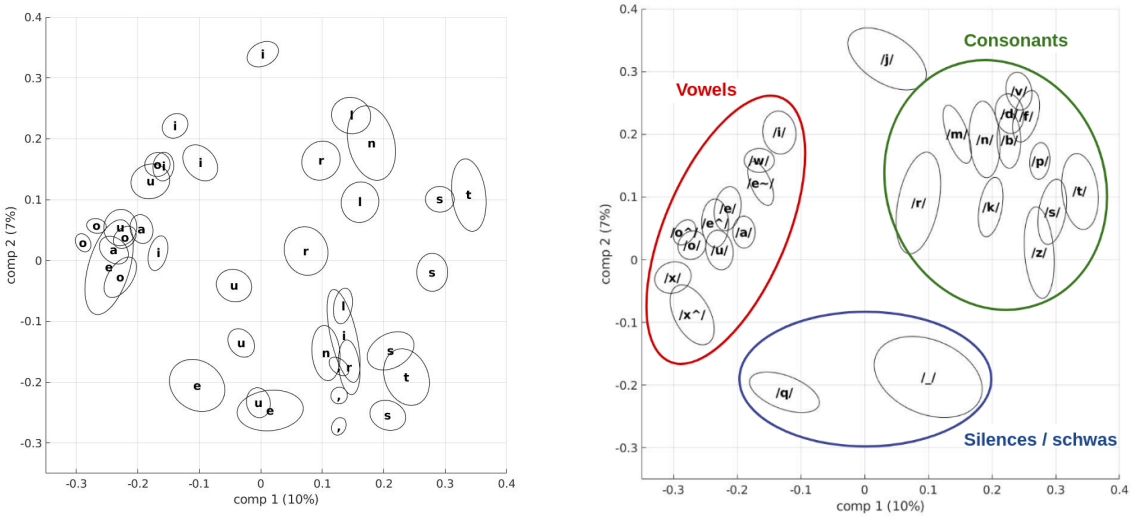
(b) F1-score by layer for Tacotron2 variants

Figure 4.3: Performance of the phonetic classification by layer. Performance is evaluated with weighted F1-score for multi-class classification (all phones, vowels and consonants), and with F1-score for bi-class classification (pauses and liaisons). Pause and liaison detection are evaluated at word boundaries.

duration computation to the output of the text encoder. Note that $\boldsymbol{TC_P}$, which implements a duration predictor, shows better predictive performance than $\boldsymbol{TC}$ in this case. Additionally, as mentioned in Section 2.2, attention maps computed by Tacotron2 sometimes show lookaheads or residual focus patterns at word boundaries. Residual focus in particular appears during the production of intra-utterance pauses on spaces, in opposition with pauses produced in accordance with punctuation marks. Residual focus is ignored by our proposed method of automatic reading of the attention map. Thus, the duration prediction is less accurate on Tacotron2 than on FastSpeech2, resulting in poorer predictive performance. This assumption will be verified in Section 4.2.3.

These results show the effect of the phonetic prediction sub-task: $\boldsymbol{FS_{\setminus phon}}$, which is the only model without phonetic prediction implemented at the output of the text encoder, shows notably worse classification performance. This effect is more visible on silence prediction. As stated in Appendix A.3, two symbols (/_/ and /__/) are used to distinguish between muted characters (with null duration) and silences (with non-null duration), respectively. As a result, when the phonetic predictor is implemented, intermediate representations are shaped to minimize the confusion between silences and muted characters at word boundaries, which increased the prediction accuracy on pauses.

The phonetic prediction F1-scores close to one at the output of the text encoder emphasizes the potential benefits of using these representations for auxiliary tasks like L2S transcriptions. As an example, the embedding space outputted by the $\boldsymbol{TC}$ encoder is showed in Fig. 4.4,



(a) By character (15 most common characters + ",")

(b) By phone (25 most common phones + /_/)

Figure 4.4: The first two components of the MDS of the embedding space outputted by the $\boldsymbol{TC}$ text encoder with orthographic input. The same embeddings are represented in both Fig. 4.4a and 4.4b, but according to their orthographic label (left), or corresponding L2S mapping (right). Ellipses indicate the distributions of each cluster along the two main axes, with one standard-deviation of amplitude.

both using orthographic inputs, and displayed with orthographic labels (Fig. 4.4a) and their L2S aligned phone labels (Fig. 4.4b). This visualization confirms the results of Perquin et al. [2020]: the embedding space is organized in unique distinct phonetic clusters that are clearly identified in Fig. 4.4b. The character inputs are projected to different locations of the latent space, depending on their corresponding phonetic output, as shown on Fig. 4.4a. Each orthographic cluster encodes one phonetic variant which can be produced from this character. The distribution along the two main axes of each ellipse encodes the early intra-phone variability computed at this stage of the process.

The performance of these pre-trained representations applied to L2S transcription were evaluated in Section 2.4.2. Predictive performance reach 99% on the test set. These results advocate for the benefits of a better understanding of latent representations that are manipulated by neural models, which enables us to design careful sub-tasks intended to enhance the performance of neural TTS.

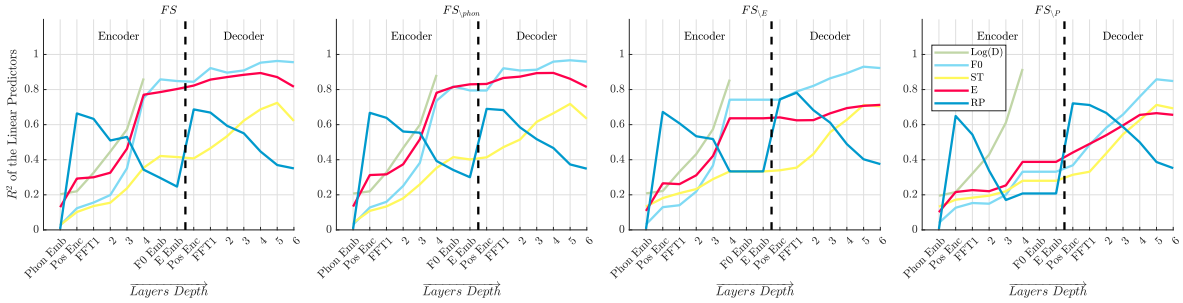
4.2.3 Supra-Segmental Features

Supra-segmental features refer (but are not exclusive) to rhythm and emphasis patterns which go beyond the local scope of phonetic identity. In that sense, supra-segmental features are involved in prosody. Contrary to segmental features, supra-segmental features are not phone-dependent: the mean of each parameter is the same for all phoneme-classes. Thus, the target of the multi-linear regression is set as the raw value of each parameter. Fig. 4.5 shows the goodness of fit of the multi-linear regressions computed by layer for Tacotron2 and FastSpeech2.

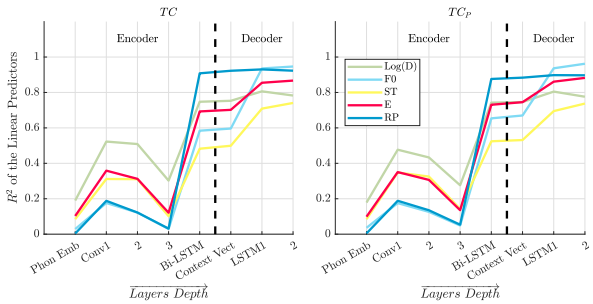
Tacotron2 and FastSpeech2 encode supra-segmental features differently. As shown in Fig. 4.5b, $\boldsymbol{TC_P}$ exhibits little changes due to the implementation of the prosodic predictors. Even in the absence of predictors, all supra-segmental features can be linearly predicted from the encoder output more accurately than segmental features. Only F0 and ST seem to benefit from the recurrent process in the decoder and reach a maximum of correlation at this stage. The first three convolutional layers introduce contextual information from adjacent phones, which helps encoding prosodic information to a certain extent. However, three convolutional layers may be excessive, as shown by the decrease of predictive performance in the third layer. This decrease of performance in the third convolutional layer is also exhibited in Fig 4.2b and Fig. 4.3b. This result indicates that two convolutional layers may be sufficient to contextualize text embeddings in TTS encoder[1]. The Bi-LSTM appears to be necessary bottleneck to select the most relevant parameters to encode in the sequence.

On the other hand, the internal representations of FastSpeech2 are very sensitive to the implemented predictors. Indeed, Fig. 4.5a shows that in the absence of prosodic predictors, $\boldsymbol{FS_{\setminus P}}$ do not encode F0 or E at the output of the text encoder, but later in the decoder. Similarly, ST, which is never explicitly targeted by a prediction task during training, only appears in the last layers of the decoder for all models. Again, FastSpeech2 higher-level decoder compared to Tacotron2 allows its encoder to compute more abstract hidden representations

[1]The presented results are computed on phonetic inputs. A wider context may be useful to model orthographic inputs.



(a) R^2 by layer for FastSpeech2 variants



(b) R^2 by layer for Tacotron2 variants

Figure 4.5: Goodness of fit of the Multi-Linear regressions by layer on supra-segmental features. The x-axis indicates the successive layers of the model (from left to right): "FFT" refers to Feed-Forward Transformer, and "Conv" to convolutional layer. Goodness of fit is expressed as the R^2 of the multi-linear regression. Acoustic features are detailed in Table 4.1.

which are gradually converted to spectral features in order to in-paint the mel-spectrogram at the decoder output.

The encoding of the relative position (RP) of the phone in the sequence shows interesting properties. FastSpeech2 implements an explicit sinusoidal positional encoding, which is summed at the beginning of the encoder and the beginning of the decoder (layer "Pos Emb" in Fig 4.5a). As shown in Fig. 4.5a, this feature fades in the encoder. This visualization reinforces the need to re-inject this positional encoding regularly in Transformer networks, which was unclear in Vaswani et al. [2017]. This finding validates the need for alternative positional encodings in deeper Transformer-based networks [Al-Rfou et al., 2019]. Without explicit encoding, this relative position is implicitly modeled by the Bi-LSTM layer of Tacotron2. Because the Tacotron2 decoder is autoregressive, it has to predict the end of the generated sequence, and modulate the voice accordingly. This causal decoding process probably benefits from having access to the current position relative to the end of the sequence. The Gate Loss Correction (GLC) introduced in Section 2.1.2 may have favored the encoding of the relative position of the phones in the input sequence, ultimately resulting in the better modeling of phrasing evaluated in Section 3.2.2. It would be interesting to apply the presented linear probing methods to a Tacotron2 without GLC to confirm this hypothesis.

4.3 Discussion of the Proposed Tracking Methodology

The proposed analysis of intermediate embeddings computed by neural TTS models shows the richness of such representations. As shown in Section 4.2.2, the set of trainable text embeddings learns to encode the variability of potential acoustic outputs from the same input symbols. First, mean acoustic features per phone class are loaded at inference time from these trained embeddings. On top of this acoustic baseline, contextualization performed in successive neural layers models the bi-directional co-articulation effects [Modarresi et al., 2004] and predicts the prosody incrementally. The tracking by layer of acoustic and prosodic features suggests that these features are incrementally built from the partial representations of the previous layers, with little to no fading, except at the very end of the decoder. In both architectures, the end layer of the decoder is passed through a fully connected layer to predict the 80 mel-coefficients of the spectrum. Thus, features not linearly encoded in the spectrogram may not be prominent in end representations.

4.3.1 Implementation of Predictive Sub-Tasks

This visualization of the evolution of spectral and prosodic features in successive layers of neural TTS is, to the best of our knowledge, the first attempt to understand the dynamics of computations learned by such models in an unsupervised way. These results mostly validate the role of the text encoder as the main contextualization tool. Although segmental features need to be refined through the decoder, their mean values are found in phone embeddings right from the start, which is enough to set the spectral targets of the decoding process.

At the output of all the Tacotron2 variant encoders, as well as $\boldsymbol{FS}$ and $\boldsymbol{FS}_{\setminus \boldsymbol{phon}}$ encoders, textual representations have already reached their maximum goodness of fit for most supra-segmental features (all except ST). Interestingly, this early encoding of F0 and E requires the implementation of prosodic predictors in FastSpeech2, but not in Tacotron2 (see model $\boldsymbol{FS}_{\setminus \boldsymbol{P}}$ in Fig. 4.5a and model $\boldsymbol{TC}$ in Fig. 4.5b). Early prosodic predictors of FastSpeech2 allow the model to anticipate prosodic variations, which seemingly helps the overall decoding process, as shown by the better perceptual scores obtained with prosodic predictors reported by Ren et al. [2021]. This difference between Tacotron2 and FastSpeech2 is likely due to the simpler structure of the Tacotron2 causal decoder, which accesses internal representations more directly than Transformer layers. With the increasing complexity of models, there is no guarantee of finding such common encoding space for all important prosodic parameters in transformer-based (or conformer-based) models. These results confirm that adding predictors forces the model to build a common space of representations, which facilitates control through linear biases. Another approach would be to use the proposed methodology to design the prediction sub-tasks in layers that already perform these tasks, to ease the training process and transform these layers into control spaces.

4.3.2 Limitations of the Linear Regression

The proposed methodology is applied to two state-of-the-art TTS architectures to emphasize that the procedure is not model dependent. As a matter of fact, similar procedures have been applied to interpret unsupervised VAE latent spaces trained on the encoding and decoding of speech spectrograms [Jacquelin et al., 2023]. The authors reported disentangled F0 and formants linear representations in the VAE-reduced space. However, the results may vary widely between models, so conclusions do not automatically transfer to other architectures. Even the addition of an auxiliary module may change how acoustic information is organized in the model, as shown by the differences between $\boldsymbol{FS}$ and $\boldsymbol{FS}_{\setminus \boldsymbol{P}}$. The addition of a utterance-wise bias like speaker or style may produce a similar effect. Nonetheless, we want to emphasize through this study that neural models should not be considered as black boxes. The proposed post-hoc analysis of latent representations enables us to interpret the model behavior based on actual knowledge of the field. We believe that understanding the models' prediction procedure is a requirement to build not only more powerful, but also more meaningful models in the future.

The proposed methodology only focused on the encoding of linear representations in intermediate embeddings. However, nothing in the models enforces the linear encoding of acoustic features. The correlations measured may be a by-product of the richness of the hidden representations computed by these neural models. To validate these correlations as causal effects from the encoded features to the generated synthesis, we designed a causal control procedure to introduce linear biases into latent representations and evaluate the effect on synthesis. This procedure is described in Chapter 5. Future work may include other probing methods. Note however that Vaidya et al. [2022] used a multilayer perceptron with one hidden layer to probe neural representations in a similar manner, but found similar results as with linear predictors.

Chapter 5

Explicit Acoustic Causal Control Through Embedding Bias

Contents

Chapter Highlights

This chapter presents how the analysis of TTS embeddings proposed in the previous chapter is **translated into explicit control** of **continuous acoustic features**, as well as the control over **phonological features** like pauses. We show that our proposed embedding bias control provides **improvements over other explicit methods**. In particular, the **combined control** of phone duration and model propensity to produce pauses offers a **more natural** control of the speaking rate.

Related contributions: [Lenglet et al., 2022b, 2023a]

This chapter extends the analysis of internal representations of neural TTS models presented in the previous chapter. Chapter 4 has highlighted in which layers of the models the selected acoustic and phonetic features were linearly encoded. The localization of features in embeddings has been evaluated through the predictive performance of the trained predictors. However, the evaluated correlations between the linear predictions from the embeddings and the measured acoustic values does not fully assess whether these are the representations used by the model to compute acoustic parameters. To assess whether the relationship between the linear representations found in embeddings and parameters produced in synthesis is causal, this chapter introduces our proposed method of linear bias computation from intermediate representations in order to control the exhibited acoustic features.

Following the additive bias framework widely used to combine embeddings from various sources in TTS [Y. Wang et al., 2018; Wu et al., 2019; Y.-J. Zhang et al., 2019], we introduce causal control as the explicit manipulation of features encoded into TTS model embeddings, by the addition of biases computed solely from the post-hoc analysis of how the learned information is structured. This causal control is inspired from work in other fields, namely neural machine translation [Bau et al., 2019] and image generation [Yang et al., 2021]. To the best of our knowledge, this approach has never been applied to TTS models. The introduction of causal biases to explicitly control acoustic and prosodic features is seen as a first step into expressive control, without the need for additional training or costly labels. In that sense, our method lies as a compromise between implicit and explicit approaches described in Section 1.2.2, introducing causal control on acoustic features computed at phone scale.

Section 5.1 describes how these linear biases, called **Embedding Biases**, are deduced from the embedding space analysis. The impact by layer of the proposed Embedding Biases is evaluated on continuous segmental and supra-segmental features in Section 5.2. The proposed causal control implicitly takes advantage of the co-variations between features learned by the models: these co-variations are compared to an equivalent disentangled explicit control in Section 5.3. The causal control is adapted to phonological control of the model propensity to produce pauses. This control is evaluated in combination with the control of duration in order to provide a more natural control of speaking rate in Section 5.4.

5.1 Embedding Bias From Latent Space Analysis

Regressions in latent spaces computed in Section 4.1 approximate any acoustic or prosodic parameter P from embeddings in the reduced space $E_L = (e_{1,L}, e_{2,L}, \ldots, e_{d,L})$ (with d being the number of reduced dimensions after parsimonious selection and L the selected layer) according to formula 4.1. This method is inspired by previous studies which showed acoustic correlations with utterance-wise representations computed by a Style Encoder [Tits et al., 2019]. Although those authors did evaluate the expressive controllability through sampling in this stylistic latent space [Tits et al., 2021], their evaluation did not take advantage of the interpretability of dimensions given by their acoustic analysis. On the contrary, we show in this section how the tracking of acoustic features through linear predictors provides explicit control mechanisms of found features.

One implication of equation 4.1 is that the direction of maximum variation of a particular continuous feature P in the reduced space, i.e. the gradient, is directly given by the regression coefficients, according to formula 5.1.

Continuous control of parameter P can then be obtained by translation along the $\nabla \hat{P}_L$ direction in the reduced space, at any layer L of the model. For ease of control, the pseudo-inverse of A_L^P, annotated $A_L^{P\dagger} = A_L^P / ||A_L^P||^2$, is used as the **embedding bias** to be added to phone or character embeddings in order to add on offset k[1] to $\hat{P}_L$, according to formula 5.2. This bias is summed to the sequence of text-embeddings at any layer of the model. Note that in order to control inference, $A_L^{P\dagger}$ has to be projected back in the model non-reduced latent space, which is made possible by the use of the MDS, which is a linear space reduction method.

$$\nabla \hat{P}_L = \begin{bmatrix} \frac{\partial f}{\partial e_{1,L}}(E_L) \\ \frac{\partial f}{\partial e_{2,L}}(E_L) \\ \vdots \\ \frac{\partial f}{\partial e_{d,L}}(E_L) \end{bmatrix} = \begin{bmatrix} \alpha_{1,L}^P \\ \alpha_{2,L}^P \\ \vdots \\ \alpha_{d,L}^P \end{bmatrix} = A_L^P \tag{5.1}$$

with $\hat{P}_L$ the approximation of P at layer L, f the linear predictor from equation 4.1, $A_L^P = (\alpha_{1,L}^P, \alpha_{2,L}^P, \ldots, \alpha_{d,L}^P)$ the regression coefficients for the parameter P along the d dimensions at layer L.

$$\hat{P}_L^{bias} = f(E_L + k \times A_L^{P\dagger}) = (E_L + k \times A_L^{P\dagger}).A_L^P + \beta_L^P = E_L.A_L^P + \beta_L^P + k \times A_L^{P\dagger}.A_L^P = \hat{P}_L + k \tag{5.2}$$

Like the linear predictors described in Section 4.1, the embedding bias $A_L^{P\dagger}$ is deduced from the statistical analysis of syntheses generated on a subset of the training corpus. This method can be applied to any continuous acoustic parameter measured on the synthetic speech produced by the model, which makes it highly versatile. Moreover, this method does not require any additional training or data.

5.2 Evaluation of the Causal Control on Continuous Features

The same models as in the previous experiment (see Section 4.2.1) are used in this section, recapped in Table 4.2. The embedding bias control is implemented on a selected set of layers in both Tacotron2 and FastSpeech architectures. For Tacotron2, three layers are selected in the encoder: after the first and third convolutional layers (Conv1 and Conv3) and after the Bi-LSTM layer. The control is also evaluated in the context vector (CV) of the decoder,

[1] k is expressed in the same unit as the predicted feature in equation 5.2.

defined as the linear combination of the embeddings computed by the text encoder weighted by their attention weights at each decoder autoregressive step. Despite some parameters being better encoded in the hidden states of the recurrent layers of the Tacotron2 decoder, the bias of recurrent layers produces instabilities, so recurrent layers were excluded from the test. For FastSpeech2, the control is evaluated after the second and fourth Feed-Forward Transformer (FFT) layers of the encoder (FFT2 and FFT4 in the encoder). In the decoder, the control is evaluated after the third, the fifth and the sixth FFT layers (FFT3, FFT5 and FFT6 in the decoder). This selection enables us to evaluate how well the control is correlated to the linearity of acoustic representations in intermediate embeddings.

5.2.1 Evaluation Procedure

To evaluate the controllability of continuous features provided by the proposed embedding bias method, linear biases are applied in order to modulate each feature in the range $[-3, +3]$ standard deviations of this parameter measured on the full ground-truth train set. The proposed embedding bias computation is based on the statistical distributions observed in latent spaces, which limits the realistic biases applicable to 3 standard deviations around the mean parameter.

This control is performed by feature, following formula 5.2[2]. Early experiments on such control of continuous acoustic features showed a saturation effect when trying to control features out of the distribution seen by the model during training. In order to avoid this saturation during the proposed evaluation, only the utterances with mean duration, fundamental frequency and energy close to the mean features are selected for further evaluation. This selection follows the same procedure as described in Section 4.1.2, but on the test set: half the test set is selected for the evaluation (1115 utterances out of 2230).

This test sub-set is generated with biases varying in the range $[-3, +3]$ (0.5 increment, ± 2.5 excluded), for each acoustic feature. Acoustic features are then measured on the biased syntheses, in order to evaluate the impact of each given bias. Despite the bias being computed only using vowel embeddings, it is summed to all the embeddings sequence. This bias performs a similar contribution to style or speaker embeddings which are generally also summed utterance-wise [Y. Wang et al., 2018; Wu et al., 2019; Y.-J. Zhang et al., 2019].

Fig. 5.1 illustrates our evaluation of the controllability. The control of F0 at layers Conv1 and Bi-LSTM of $\boldsymbol{TC_P}$ is taken as an example: for both layers, F0 is measured on the syntheses for each vowel of the test set and for each bias in the range $[-3, +3]$. Both target and achieved biases are expressed relative to the standard deviation of F0 in the ground truth. For ease of interpretation, the achieved bias is expressed as a function of the target through a sigmoid regression. This regression enables us to measure two aspects:

1. **The goodness of fit of the sigmoid regression ($\mathbf{R^2}$)** measures the variability around the mean control achieved by the proposed method. This variability should be minimal to ensure an accurate control. The sigmoid regression was preferred over simpler lin-

[2] k is computed by feature to account for $[-3, +3]$ standard deviations of the controlled feature.

ear regression to take potential saturation effects into account, as illustrated in Fig. 5.1 (Layer: Bi-LSTM). In this example, the sigmoid follows closely the achieved bias, resulting in R^2 close to 1 (0.93). On the contrary, the bias has close to no effect in layer Conv1: as a result, the residual error of the sigmoid is equivalent to the variance of the data, so R^2 is close to 0 (0.02).

2. **The control range achieved:** the proposed method should reach a control range as close as possible to the target [-3, +3] standard deviations range. This range is given by the extreme values of the sigmoid regression. In practice, our observations of the bias effect showed a symmetrical effect when increasing or decreasing a parameter. Thus, ranges are expressed as absolute values.

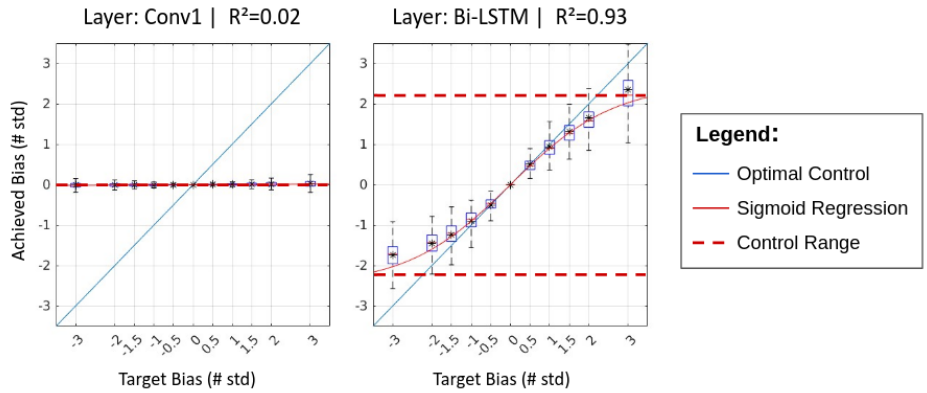
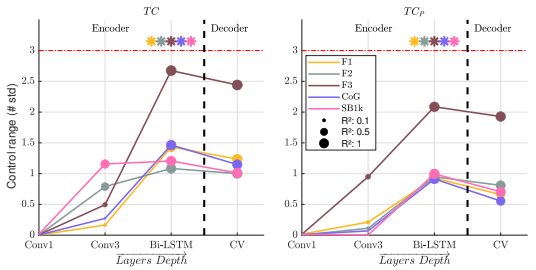


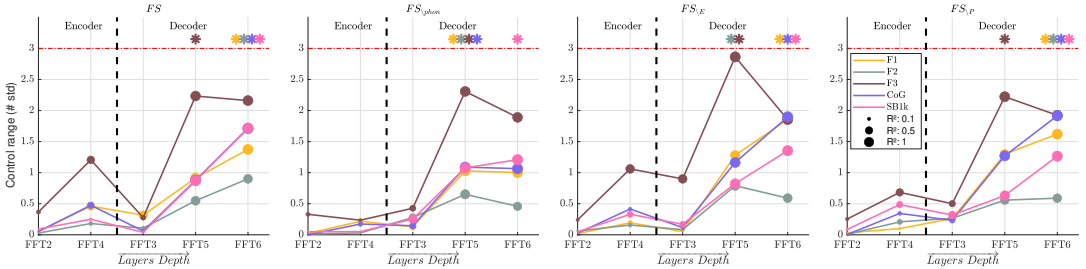
Figure 5.1: Illustration of the controllability evaluation of F0 for 2 layers of $\boldsymbol{TC_P}$: Conv1 and Bi-LSTM.

5.2.2 Control Performance

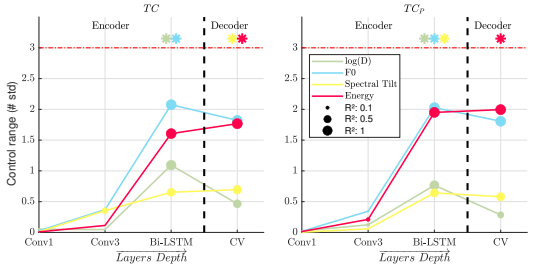
Fig. 5.2a-5.2d show the evaluation of the controllability of all measured continuous features for Tacotron2 and FastSpeech2 variants. The controllability is mostly correlated to the localization of linearly encoded features from Fig. 4.2 and Fig. 4.5. Regardless of the presence of prosodic predictors, Tacotron2 shows its best controllability performance (both for R^2 and control range) after the Bi-LSTM layer for all features but E. The addition of the bias to the output of the encoder allows the attention mechanism to take this bias into account, whereas adding the bias to the context vector by-passes the attention, which explains the drop in performance of the duration control in the context vector.



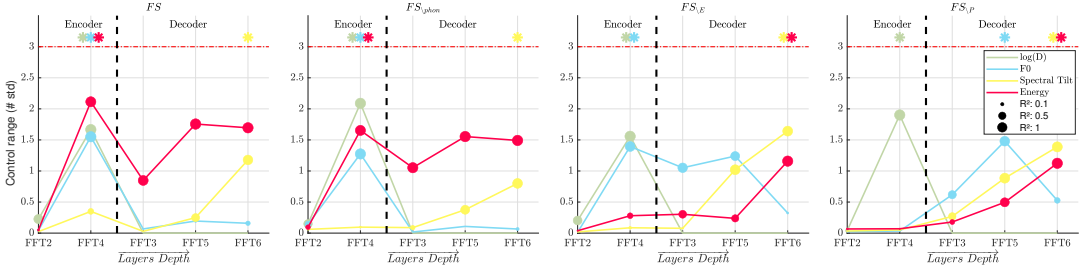
(a) Controllability of segmental features for Tacotron2 variants



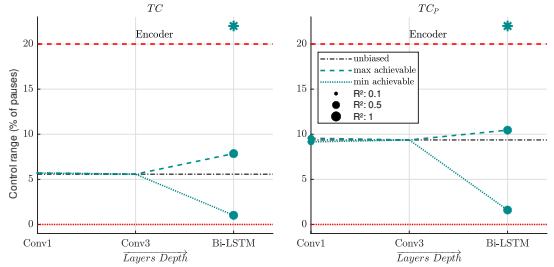
(b) Controllability of segmental features for FastSpeech2 variants



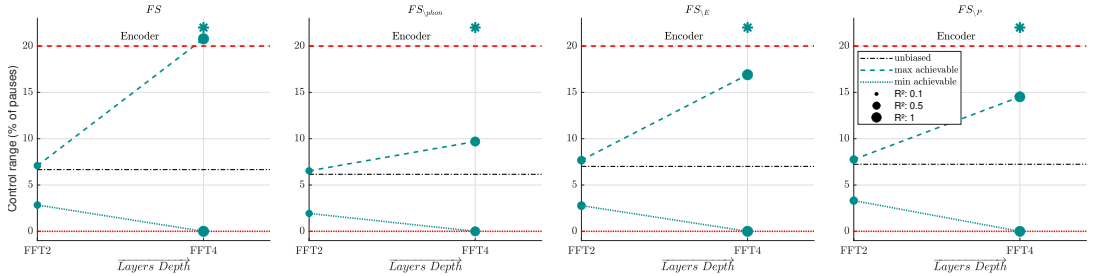
(c) Controllability of supra-segmental features for Tacotron2 variants



(d) Controllability of supra-segmental features for FastSpeech2 variants



(e) Control range of pauses for Tacotron2 variants



(f) Control range of pauses for FastSpeech2 variants

Figure 5.2: Controllability of encoded features by the proposed embedding bias method. * indicates the layer with the best controllability.

FastSpeech2 controllability results are more distributed across layers. With prosodic predictors, ***FS***, $\boldsymbol{FS_{\setminus E}}$ and $\boldsymbol{FS_{\setminus phon}}$ show their best controllability of F0 and duration at the output of the encoder (Fig. 5.2d). Surprisingly, despite the presence of the energy predictor, the energy bias has close to no effect on the control of energy at the output of the encoder for $\boldsymbol{FS_{\setminus E}}$. Instead, the energy is controllable closer to the output of the decoder, similar to $\boldsymbol{FS_{\setminus P}}$. In the absence of the pitch predictor, the control of F0 through embedding biases also appears closer to the decoder output. Similarly to Tacotron2, segmental features are controllable closer to the output of the decoder (Fig. 5.2b).

The achievable control range varies depending on the model and the controlled feature. Fig. 5.2a-5.2b show that segmental features are generally less controllable, with ranges varying within [0.5, 1.5] standard deviations for Tacotron2 (***TC*** performs better than $\boldsymbol{TC_P}$ in that regard), and [0.5, 2] for FastSpeech2. Only F3 exceeds 2 standard deviations for ***TC*** and FastSpeech2 variants. Pitch and energy on the other hand achieve [1.5, 2] at best for both architectures (Fig. 5.2c-5.2d). FastSpeech2 shows better performances for duration, thanks to the explicit duration predictor. The attention mechanism of Tacotron2 is less robust to the proposed control. Table 5.1 expresses the ranges of standard deviations into absolute control ranges.

Fig 5.3 illustrates the control of F0 on an example for ***FS***. This control is achieved at the output of the text encoder, which shows the best performance for F0 control in this model. This figure shows the three main effects of the proposed control: First, at the utterance-level, the shifting of the F0 trajectory is symmetrical, but does not achieve the target modification imposed by the bias: ±2 std should modify the mean F0 by ±5.16 st. In practice, the bias shifts the F0 by ∼3 st. Second, local saturation happens when the unbiased F0 in the synthesis already reaches extrema of the training distribution, like in [tut] (maximum) or the end syllable of [mizerablq] (minimum). Finally, the F0 contour may be modified by the bias, as illustrated by the falling pitch at the end of utterance for -2 std compared to the rising pitch in unbiased synthesis.

Feature	**Unit**	**FastSpeech2**				**Tacotron2**	
		FS	$\boldsymbol{FS_{\setminus phon}}$	$\boldsymbol{FS_{\setminus E}}$	$\boldsymbol{FS_{\setminus P}}$	***TC***	$\boldsymbol{TC_P}$
F1	st	±2.78	±2.09	±3.76	±3.29	±2.91	±1.89
F2	st	±1.99	±1.44	±1.75	±1.31	±2.39	±2.10
F3	st	±2.23	±2.31	±2.86	±2.22	±2.67	±2.09
CoG	st	±4.84	±3.08	±5.37	±5.43	±4.13	±2.57
SB1k	dB	±5.55	±3.90	±4.35	±4.09	±3.90	±3.22
D	Elongation Coef	0.76-1.31	0.71-1.40	0.78-1.29	0.73-1.36	0.84-1.19	0.88-1.13
F0	st	±4.00	±3.28	±3.59	±3.84	±5.37	±5.24
E	dB	±4.35	±3.40	±2.39	±2.31	±3.65	±4.13
ST	dB/octave	±0.89	±0.61	±1.24	±1.05	±0.52	±0.48

Table 5.1: Absolute Control Range of Acoustic Features, measured on the best control layer by model (as indicated by * on Fig. 5.2). st stands for Semitones.

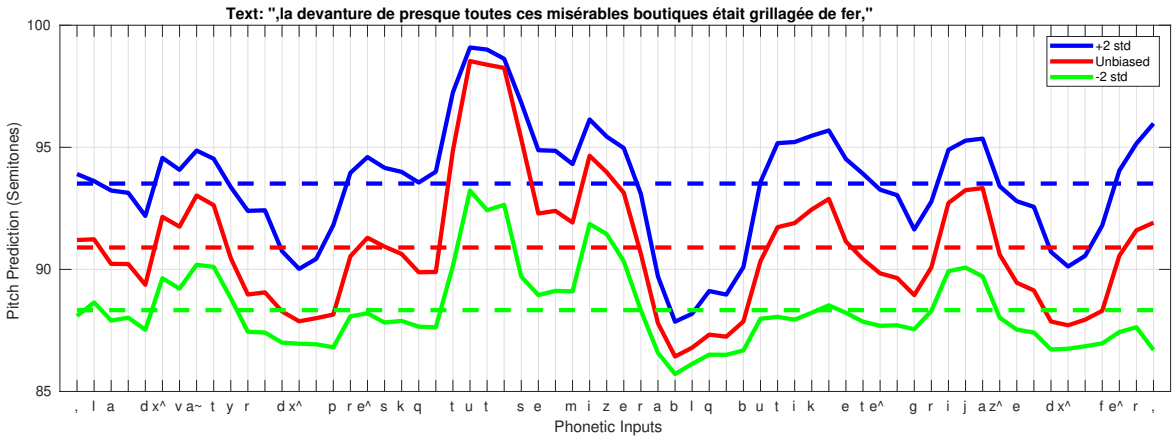


Figure 5.3: Illustration of F0 contours predicted on biased syntheses for ***FS*** at the output of the text encoder. The pitch predictor of FastSpeech2 predicts pitch values for every input symbol, regardless of voiced/unvoiced phones. Dotted lines indicates mean F0 by condition.

Despite resisting the target modification imposed by the bias, saturation and changes in pitch contour may contribute to speech naturalness. In real-life scenario, speakers are unlikely to entirely shift their pitch contour when speaking with higher or lower pitch, for physiological and communicative reasons. Speakers are limited in their range of pitch productions, which was learned by the models and is reflected through local saturation. Also, mean pitch modifications contribute to communicative purposes, either conscious or unconscious: lower pitch contributes to attractiveness [Feinberg et al., 2005] and perceived dominance [Puts et al., 2006]. On the other hand, higher pitch shows more nervousness [Apple et al., 1979]. When recording the audiobooks, the speaker simulates these communicative intents and the corresponding contour variations. Neural models can learn these covariations through statistical learning, which are then applied when biasing the internal embeddings. Covariations learned by neural models are further explored in Section 5.3.1.

The more mitigated performances of segmental features control compared to supra-segmental features may be explained by the nature of the computed biases. Biases are computed regardless of the phone identity, and summed regardless of the embedding sequence. This method relies on the assumption that all phone embeddings encode segmental features in the same direction of the latent space. However, limited predictive performance of linear predictors of segmental features (Fig. 4.2) compared to supra-segmental features (Fig. 4.5) may indicate that this direction is not unique, and should be adapted to each phone. On the contrary, supra-segmental features are shared across phones, resulting in better performance of the shared bias.

5.2.3 Mismatch between Linear Biases and Acoustic Effects

Although we have seen that the addition of a linear bias can control to a certain extent the generated acoustic features, this control is contested by the models which only partially apply the target modifications. Our hypothesis is that the discrepancy between the target bias and the measured acoustic modification is due to the multitude of representations found in the latent space to encode the same feature. Even though the biases don't precisely match the target, they still produce consistent modifications, especially for the best control layers where the goodness of fit is close to one. This may show that the proposed procedure did not manage to isolate the dimension which encodes the desired feature, but rather was impacted by the multitude of covariations found between the internal dimensions of neural latent spaces. As a result, part of the bias does contribute to the model computation of the target feature, but the rest is summed to different dimensions, resulting in noise added to the other features. Two observations support this hypothesis:

1. Because the proposed explicit control mechanism relies on the addition of linear biases, a trivial adaptation of the procedure to compensate for the mismatch between target and measured modifications is the addition of a multiplying factor to modify the embedding bias amplitude. We used this adaptation to ensure compliance with changes in the target duration modifications in Lenglet et al. [2022b]. This trivial multiplying factor (set as the ratio between the target and the obtained modification measured in Fig 5.2c-5.2d) does ensure a match between the target and the acoustic modification.

2. Inversely, despite the high correlation of F0 found in the decoder of ***FS*** (Fig 4.5a), the control of F0 has no effect after the F0 predictor (Fig 5.2d). This emphasizes that several representations of the same acoustic feature may coexist inside latent spaces. Thus, finding one linear representation does not ensure that this representation is used as it is by the model. If several linear representations co-exist, the resulting embedding bias combines these contributions, instead of focusing on the meaningful one.

This concern is the reason why we tried to refine the embedding bias computation procedure with dimensional reduction and parsimonious dimension selection described in Section 4.1.2 compared to the procedure we used in Lenglet et al. [2022b]. Through this refining, the ratio between the achieved control and the target duration bias was increased from 0.43 to 0.56 for ***FS*** (resp. from 0.34 to 0.36 for ***TC***) compared to our previous experiment [Lenglet et al., 2022b]. This improvement validates the refining of the computation procedure, but advocates for an even greater reduction in the number of dimensions which should be the focus of future work.

The addition of a correction multiplying factor method is not ideal: this coefficient is inherently model/layer/feature specific, since it compensates for the flaws of the proposed method in finding the actual direction of encoding of the said feature. However, it shows how the proposed causal control may be adapted to maximize the control range achievable. Additionally, we emphasize that the proposed method does not require any additional training of the models, or extensive datasets to maximize the variability of features during training. Though, note that a wider range of variation of measured acoustic features is expected to contribute to wider control ranges.

5.3 Embedding Bias VS Explicit Control

Computing biases based on statistical learning performed by neural models provides some interesting perspectives in comparison with explicit control mechanisms. With the introduction of linear biases into intermediate embeddings computed by TTS models, the proposed method stands out from the disentangled control paradigm advocated by numerous TTS studies [Mohan et al., 2021; Raitio et al., 2020; Y.-J. Zhang et al., 2019]. On the contrary, linear biases take advantage of correlations between acoustic features learned during training. Thus, the control of one parameter is expected to impact other acoustic features. Section 5.3.1 evaluates these covariations.

In addition, explicit control as provided by models like FastSpeech2 or Ctrl-P [Mohan et al., 2021] enable control of prosodic features at the phone level at inference. This is very useful to precisely control the output synthesis when the human operator has his own prosodic target in mind, either given by a reference speech to mimic or by a hand-in-the-loop procedure [Koch et al., 2022]. Local control is also useful to emphasize target words or parts of sentences [Joly et al., 2023]. However, in most cases, less supervised control methods are preferred in order to ease the synthesis pipeline. As a proof of concept, Large Language Models (LLM) were tested as replacement of humans explicit control of prosodic features with a FastSpeech2 model by Sigurgeirsson and King [2023]. Despite the promising performance of the LLM on the prediction of prosodic contours to mimic expressive styles, the lack of understanding of the predictions made by such models is a hindrance to their implementation in real-life applications.

So, in absence of prosodic contours at phone level, explicit prosodic control usually applies the same modification throughout the entire sequence to bias. While this ensures that the mean of the controlled feature is scaled according to the target, uniform modifications of features may degrade speech naturalness. As an example, variation of phone duration with speaking rate depends on phoneme and position in the sentence [Nick Campbell, 1992], in opposition with uniform duration modifications. This section explores two hypotheses that might favor the natural control with embedding biases compared to explicit control:

H1: Neural TTS models learn natural covariations between acoustic features from the training corpus. These covariations are found in the encoding of acoustic features in the embeddings. This hypothesis is explored in sub-section 5.3.1.

H2: The discrepancy between the acoustic target control with linear biases and the achieved effect is due to the model having statistically learned plausible modifications from natural speech, resulting in more careful productions when biased. This hypothesis is explored in sub-section 5.3.2.

5.3.1 Covariations in Intermediate Embeddings

Embedding biases are computed by statistical analysis of intermediate embeddings. As stated in Section 4.1.2, the proposed method of acoustic tracking in intermediate embeddings maximizes the accuracy of linear predictions of each individual acoustic feature independently of the others. As a result, gradient directions computed by this method for each acoustic feature are not constrained to be orthogonal. Instead, the directions of maximum variation indicate how linear representations of acoustic features co-vary in embedding spaces. These empirical covariations are learned by the models as regularities found in the training corpus. Thus, following these statistical covariations may be beneficial for a more natural control of the synthesis.

Covariations of internal representations can be predicted in any layer of any model through the proposed analysis. For the sake of simplicity, we focus this analysis on the two vanilla architectures ***TC*** and ***FS***, and on the layers that shows the best controllability, chosen as follows. The output of the text-encoder[3] shows the best potential for supra-segmental features (as well as segmental features for Tacotron2). Despite the later encoding of segmental features in FastSpeech2, we consider prosodic parameters (fundamental frequency, duration and energy) to have a wider impact on the speech perception. Thus the output of the text encoder is the best layer to implement a combined control.

Covariations of acoustic features in these embedding spaces are predicted through mutual orthogonal projections of embedding biases, reported in Fig 5.4. First, the angle between each pair of embedding biases $A_L^{P\dagger}$ is computed. Since embedding biases have different magnitudes

[3]Bi-LSTM for Tacotron2, and FFT4 for FastSpeech2.

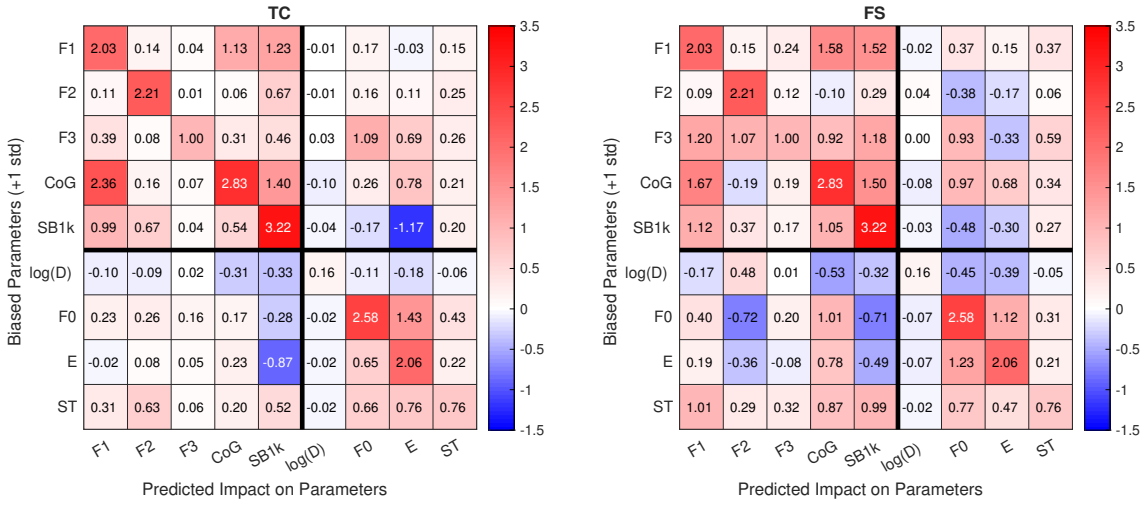


Figure 5.4: Covariations predicted from the output of the text encoder for ***TC*** (left) and ***FS*** (right). Predicted covariations on impacted parameters (x-axis) indicate predicted feature modifications for a bias of +1 standard deviation of control parameters (y-axis). Reported numbers of resulting features modifications are expressed in the units given in Table 5.1 for each column.

depending on the variance observed in the corpus for each parameter, mutual orthogonal projections illustrate the asymmetrical effect of the control of one feature on the others. Individual controlled features are given on the y-axis. Then, the x-axis illustrates the predicted modification of all measured features, for an increase of one standard deviation of the controlled feature. Predicted impact on all features are reported following their own unit, given in Table 5.1. By construction, the diagonal reports the standard deviation of all measured features. As an example, the first line of the left panel of Fig 5.4 (***TC***) shows that for an increase of one standard deviation of F1, we predict a variation of 2.03 st on F1, 0.14 st on F2, and -0.03 dB on energy, etc.

TC and ***FS*** show consistent covariations prediction of supra-segmental features (lower-right area): 1) An increase of log(D) (i.e. an decrease of the speaking rate) slightly reduces F0 and E. 2) F0 and E vary cojointly. 3) Increase of ST also raises F0 and E, as well as F1, simulating an increase of vocal effort [Liénard & Di Benedetto, 1999]. Even though segmental features are not fully encoded yet, in particular for ***FS***, covariations of segmental features (top-left area) already indicate the joint modifications of F1, CoG and SB1k, which was expected from a signal processing perspective[4]. Covariations between segmental and supra-segmental features are less clear (top-right and down-left areas). Notably, SB1k and E vary in opposite directions. SB1k control tends to reduce the overall energy in the synthesized spectrum. The target balance is seemingly achieved through further reduction of one frequency band compared to the other, instead of an increase of energy in low or high frequencies. Variations of spectral balance may be rare in the training corpus, which encourages the models to rely on variations of energy instead.

As a general observation, ***TC*** shows less covariation than ***FS***. As seen in Fig 4.2, segmental linear representations modeled at the output of the encoder better correlate with acoustic measurements for ***TC*** than for ***FS***. ***FS*** embeddings are computed in a smaller number of dimensions than ***TC*** (256 compared to 512). This limited memory size may enforce ***FS*** to encode diverse acoustic features on less dimensions, which would result in more covariations. The training of several variants of models with various size of internal representations should give better insights on this hypothesis. Because of time limitations, this must be explored in future work.

We emphasize that the covariations shown in Fig 5.4 are predicted from the proposed analysis of internal spaces, and not measured on the audio output. Nonetheless, these predictions confirm **H1**: directions of maximum variations indicate common (or opposite) variations of features which match covariations expected from natural speech. These findings are promising for the integration of more natural control mechanisms into neural TTS. The evaluation of these covariations on the audio output will be explored in future work.

[4]Increase of F1 shifts spectral energy toward higher frequencies, which increases CoG. This modification is not enough to push formants across the 1 kHz limit though. Mean F1 is 107 st, compared to 1 kHz $\approx$ 120 st.

5.3.2 Non-Linear Duration Control

The non-linearity of the control of the proposed embedding bias is illustrated on F0 in Fig. 5.3. Our hypothesis **H2** states that this non-linearity may be beneficial for the naturalness of the synthetic voice. This sub-section evaluates this hypothesis. Duration is taken as an example. Results presented in this sub-section are taken from: Lenglet et al. [2022b]. This paper is attached at the end of this manuscript.

5.3.2.1 Experimental Setup

This experiment specifically evaluates the control given by the duration embedding bias, expressed in log-duration. Hence the addition of a bias in the log domain is equivalent to applying a multiplying factor on phone duration. In both cases, this embedding bias is applied at the output of the text encoder of the two models ***TC*** and ***FS***. This control is compared to two explicit duration control mechanisms taken as baselines: 1) The explicit duration control provided by FastSpeech2, annotated $\boldsymbol{FS_C}$ in the following, and 2) a simple linear time-interpolation of the mel-spectrogram output of the unbiased synthetic model to change the full duration of the signal before feeding it to the neural vocoder. This method is annotated ***Stretching***. In both baselines, a similar modification of duration is applied on all phones, but $\boldsymbol{FS_C}$ has the chance to make some acoustic modifications through the decoding process. In this section, $\boldsymbol{TC_B}$ and $\boldsymbol{FS_B}$ refer to the duration embedding bias control for ***TC*** and ***FS*** respectively.

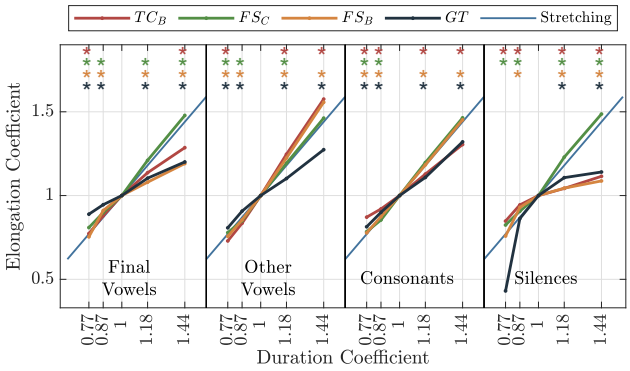
In order to evaluate the local differences between the proposed embedding bias control and explicit methods, the test corpus[5] is synthesized with 4 duration coefficients, chosen to be representative of the phone rate distribution of the training dataset. These coefficients $m_i = \{0.77, 0.87, 1.18, 1.44\}$ are chosen to reach $i = \{+2, +1, -1, -2\}$ standard deviation around the mean phone rate, respectively. As discussed in Section 5.2.3, a multiplying factor is used to ensure that target duration modifications are reached with the embedding bias method (2.94 and 2.33 for $\boldsymbol{TC_B}$ and $\boldsymbol{FS_B}$ respectively).

5.3.2.2 Duration Modification By Phone Class

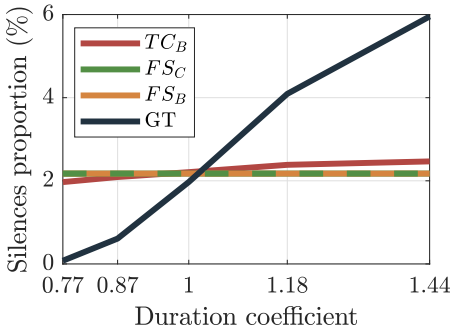
For each synthesized signal with a given duration coefficient, the duration of each synthesized phone is measured by the method described in Section 2.2.2. This duration is then divided by the mean duration of its phone class synthesized with the same model without duration control, to provide an elongation coefficient. Fig. 5.5a displays the average elongation coefficient per duration coefficient, model, and phone class. Final vowels are vowels just preceding a silence in the audio signal. For each phone class, the diagonal corresponds to the ***Stretching*** condition, where the elongation coefficient equals the duration coefficient. The red, green and yellow curves correspond to $\boldsymbol{TC_B}$, $\boldsymbol{FS_C}$, $\boldsymbol{FS_B}$, respectively. Moreover, average phone elongation coefficients were also calculated on the ground truth training corpus (***GT***) and reported in dark blue[6].

[5]Same test set as the previous experiment: see Section 4.2.1

[6]Relative speaking rates by utterance were computed on the Ground-Truth training database. Utterances with relative speaking rates in a window of 0.05 around the coefficients [0.77, 0.87, 1, 1.18, 1.44] were selected



(a) Elongation coefficient is the mean phone elongation compared to the unbiased voice. * indicates a significant difference with stretching.



(b) Silences proportion is the ratio between the number of silences in the audio signal and number of phones in the text input.

Figure 5.5: Impact of duration control for each model and ***GT***.

A Kruskal-Wallis rank-sum test performed on the per-phone elongation coefficients showed a significant effect of both phone class and duration control ($p < 0.01$). A post-hoc Wilcoxon rank-sum test then assessed for each phone class and duration coefficient whether each method significantly differs from the ***Stretching*** conditions. Significance ($p < 0.01$) is displayed by coloured stars above each data point. Fig. 5.5b shows the ratio between the number of pauses longer than 30 ms in the audio signal and the number of phones in the text input for each duration coefficient on ***TC_B***, ***FS_C***, ***FS_B*** and ***GT*** (by nature, this ratio do not vary with duration control for ***FS_C*** and ***Stretching***).

Concerning elongation coefficients (Fig. 5.5a), ***FS_C*** follows the diagonal: as expected, frames are linearly duplicated through duration control for any class of phones. On the contrary, ***GT*** data displays non-linear behaviors that are consistent with Nick Campbell [1992] findings. Looking first at slower speaking rates ($m_i > 1$), ***GT*** displays a saturation for final vowels and silences whose mean durations are already large for average speaking rate (125 ms and 213 ms, respectively) and therefore weakly lengthened as the speaking rate decreases. This behavior has been learned by ***TC_B*** and ***FS_B***. This validates **H2**: the discrepancy between the target bias and the achieved elongation shows that embedding bias does not allow to extrapolate to unseen phone lengths, ultimately resulting in saturation of duration which mimics the Ground-Truth.

Regarding other vowels and all consonants, ***GT*** shows a linear lengthening with duration increase but to a lesser extent than ***Stretching***. This is compensated by the introduction of pauses in the ***GT*** signals: Fig. 5.5b displays three times more pauses in ***GT*** when the speaking rate is 1.44 times slower. Conversely, ***FS_B*** does not generate any additional pauses in the signal, and the effect is negligible for ***TC_B***. Alternatively, both models compensate by expanding the vowels longer than the stretch (Fig 5.5a). On consonants, ***TC_B*** seems to have learned ***GT*** behavior, wile ***FS_B*** follows the ***Stretching*** trend.

to compute the duration by phone reported in this sub-section.

Looking now at higher speaker rates ($m_i < 1$), ***GT*** final vowels are preserved while silences are dramatically shortened or deleted (Fig. 5.5b). This behavior was not replicated by any model. For other vowels and consonants, ***GT*** and all models follow a linear shortening of phones matching ***Stretching***.

5.3.3 Discussion: Promising Performance of the Embedding Bias Control

Globally, ***GT*** duration modification by the talker is mainly performed with pause addition and deletion, that are hardly managed by the embedding bias-controlled models. Regarding the observed non-linearity per class of phones, $\boldsymbol{TC_B}$ follows best the ***GT*** behaviors, even though it compensates for the lack of pause addition by vowel lengthening. Both $\boldsymbol{TC_B}$ and $\boldsymbol{FS_B}$ follows the saturation of final vowels and pauses that are imposed by the data distribution, but $\boldsymbol{FS_B}$ mainly follows the ***Stretching*** behavior otherwise.

These results confirm that non-linearities in duration modifications found in natural speech are better modeled by the proposed embedding bias control than by explicit control. Despite the embedding bias being computed on vowel embeddings, applying the bias to every embeddings at the output of the text encoder produces the desired effect on other phone classes, as illustrated on consonants and silences in Fig 5.5a. Perceptual evaluations were performed in Lenglet et al. [2022b] to asses the distinctiveness of the proposed control. These subjective evaluations will be discussed in Section 5.4.2.

The possibility to add or remove pauses while modifying the speaking rate appears essential in order to model the natural behavior of speech. However, if a linear representation of the duration is encoded, as proved by the duration control of existing pauses provided by the proposed bias (Fig 5.5a), increasing this encoded duration does not trigger the addition of new pauses (Fig 5.5b). Moreover, preliminary experiments showed that taking duration of pauses into account when computing the duration embedding bias did not introduce more control over pause generation. This suggests that a distinct categorical pause addition/deletion trigger could also be encoded in the latent space, to indicate to the duration predictor or the attention layer whether the encoded duration should be produced or not. To explore this hypothesis, we extended the proposed procedure to track categorical representations with multi-linear predictors. The proposed adaptation is discussed in Section 5.4.

5.4 Evaluation of Categorical Control

The linear bias approach can also be adapted to discrete phonological control. As shown in Section 4.2.2.2, linear classifiers by LDA can be trained to find linear representations of classes into latent embedding spaces. In this case, linear classifiers by LDA compute the hyperplanes that best distinguish between the classes, either in a bi-class setup (pause VS absence of pause, and liaison VS absence of liaison) or with multiple classes (phone classes). Multi-class setup is not studied here, since we did not find any use for phone conversion with embedding bias.

In case of bi-class classification, only one hyperplane is computed, whose coordinates are given in formula 5.3. The vector normal to this hyperplane indicates the direction of the latent space which best encodes the switch between these two categories. The embedding bias is then defined as the vector N_L^C normal to this hyperplane, normalized so that the amplitude of the categorical embedding bias matches the differences between the barycenters of the two categories, following formula 5.4. The categorical embedding bias $\hat{N_L^C}$ can then be added to embeddings computed by the TTS models to modify the proportion of classes C_1 and C_2.

$$\mathcal{H}_L^C : \sum_{i=1}^{D} n_{i,L}^C . e_{i,L} + \gamma_L^C = 0 \tag{5.3}$$

with $\mathcal{H}_L^C$ the hyperplane that best distinguishes the two classes of C in layer L, $N_L^C = (n_{1,L}^C, n_{2,L}^C, \ldots, n_{d,L}^C)$ the vector normal to this hyperplane, and γ_L^C the intercept. D is the number of dimensions in layer L.

$$\hat{N_L^C} = (\mu_L^{C_1} - \mu_L^{C_2}) \times N_L^C / ||N_L^C||^2 \tag{5.4}$$

with $\hat{N_L^C}$ the normalized vector normal N_L^C, $\mu_L^{C_1}$ and $\mu_L^{C_2}$ the barycenters in layer L of classes C_1 and C_2 respectively.

5.4.1 The Case of Pauses

Any of the bi-class phonological features found in Section 4.2.2.2 might be controlled with this proposed adaptation. The model tendency to produce pauses at word boundaries is taken as an example of such control. As seen in Fig 5.5b, all TTS models fail to reproduce natural voice pause variations when modifying the speaking rate. This categorical control could complement the duration control to produce more natural variations of speaking rate.

Because the presence of pauses is determined by the duration predicted for characters at word boundaries (spaces and punctuation marks are preserved with phone-input), the control is only applied in the encoder layers. For the same reason, the bias is only added to spaces and punctuation marks symbols, instead of utterance-wise for continuous control.

5.4.1.1 Calibration Phase

To mimic the natural voice tendency to produce silences, the proportion of produced silences across potential silences localization in the ground-truth was evaluated by speaking rate. This proportion is illustrated in Fig 5.6a. The proportion of pauses in $\boldsymbol{GT}$ can be approximated by a linear regression in the range of duration coefficients [0.79; 1.37]. We infer the empirical proportion of pauses by elongation coefficient from this regression, following the formula 5.5.

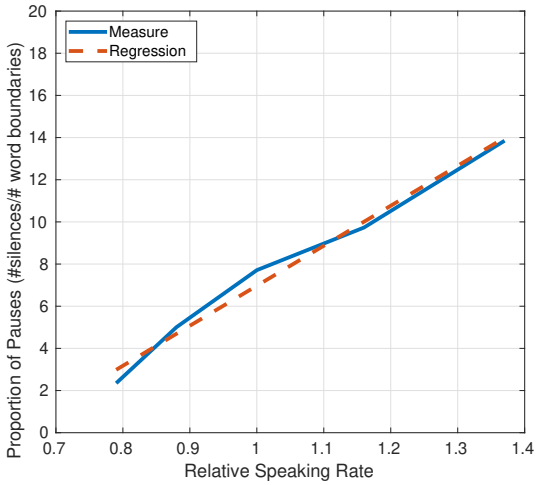
$$Pause^{\%} = \mathbf{18.98} \times \mathcal{D} \mathbf{\ -\ 12.01} \tag{5.5}$$

with $Pause^{\%}$ the percentage of pauses achieved among all possible word boundaries, and $\mathcal{D}$ the duration coefficient. Bold coefficients are inferred from the regression on ***GT*** recordings.

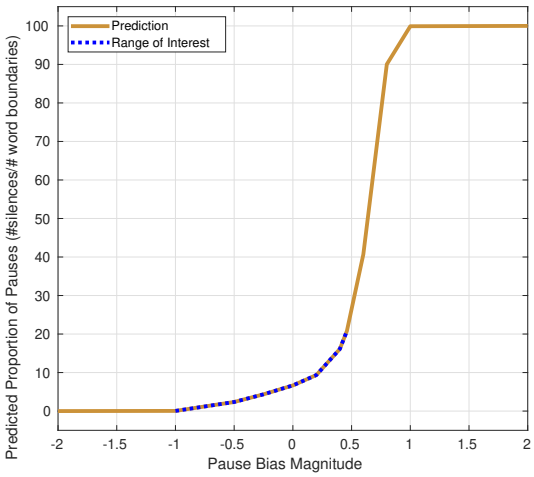
We set the observed range [0%; 20%] on the training dataset between the fastest and slowest utterances as a target for the evaluation of the control of pauses. Thus, the pause control was evaluated as the linear bias ability to vary this proportion between 0% and 20%.

5.4.1.2 Pauses Control by Linear Biases

In order to evaluate the controllability of pauses, synthesis is performed with pause embedding bias magnitudes varying in the range [-2; 2]. This range was empirically chosen to be large enough to ensure the full transfer of either class to the other side of the hyperplane computed by the bi-class LDA classifier. Predicted proportion of pauses in synthesis is computed from the analysis of distances between embeddings and the LDA hyperplane in latent spaces. Fig 5.6b illustrates this prediction for ***FS*** at the output of the text encoder (layer FFT4). Linear biases are able to extrapolate the percentage of achieved pauses above 20%, but this goes beyond the range that corresponds to natural variations in speech.



(a) Proportion of pauses achieved among all word boundaries in the **Ground-Truth** recording. Regression coefficients are given in formula 5.5.



(b) Proportion of pauses **predicted from distribution of embeddings in latent spaces**. The output of the text encoder (layer FFT4) of ***FS*** is illustrated as an example.

Figure 5.6: Calibration of the Pause Embedding Bias

The controllability of pauses is evaluated in Fig. 5.2e-5.2f, as the range of variation achieved by synthetic models for a target proportion of silences in the range [0%; 20%]. Most models show limited controllability compared to predictions. FastSpeech2 show greater controllability performances than Tacotron2, with ***FS*** achieving the [0%; 20%] control range target. This result emphasizes again the robustness of the duration prediction layer to linear control, compared to the attention mechanism of Tacotron2. The phonetic prediction layer helps to model this behavior, as showed by the poorer performances of $\boldsymbol{FS}_{\backslash \boldsymbol{phon}}$. This drop in performances was anticipated by the worse classification accuracy of pauses evaluated on $\boldsymbol{FS}_{\backslash \boldsymbol{phon}}$ in Section 4.2.2.2.

Although this evaluation of the controllability of pauses was run in isolation, the main benefit of the control of pause addition/deletion is expected in combination with the control of duration. Indeed, the combination of the continuous bias to modify duration of phones in the sequence with the pause bias to produce more or less pauses is expected to mitigate the main flaw of TTS models when controlling the speaking rate. In order to combine both contributions, an abacus is derived from the Figures 5.6a and 5.6b. This abacus is used to estimate the pause bias magnitude to combine with the target duration coefficient in order to generate the appropriate proportion of pauses in synthesis. This combined control is evaluated in Section 5.4.2.

5.4.2 Speaking Rate Modeling

The speaking rate control with combined duration and pause bias is evaluated on the variant of each Tacotron2 and FastSpeech2 which allows the best control of proportion of pauses: ***FS*** and $\boldsymbol{TC_P}$. $\boldsymbol{TC_P}$ showed lower performance in duration control than ***TC***, but ***TC*** reduces the range of the pause control too much. In both cases, the duration and the pause embedding biases are added at the output of the encoder, right before the duration predictor in ***FS*** and before the attention layer in $\boldsymbol{TC_P}$. Two abacuses are recorded through the calibration phase described in Section 5.4.1.2, to predict the magnitude of the pause embedding bias according to the expected pauses proportion by duration coefficient in the ***GT***. These abacuses are given in Appendix E.

Fig 5.7 illustrates the comparison between the ***Stretching*** and the combined embedding bias control of duration and pauses. The utterance is ",son audace ne l'eût pas abandonné devant un tribunal ordinaire;", and the target duration coefficient is 1.38. The embedding bias method introduces a clear pause of 200ms between the verb "eût pas abandonné" and the complement of place "devant...". Words are also better segmented, with less residual energy in word boundaries. The absence of pauses is compensated in the ***Stretching*** by an excessive elongation of the words.

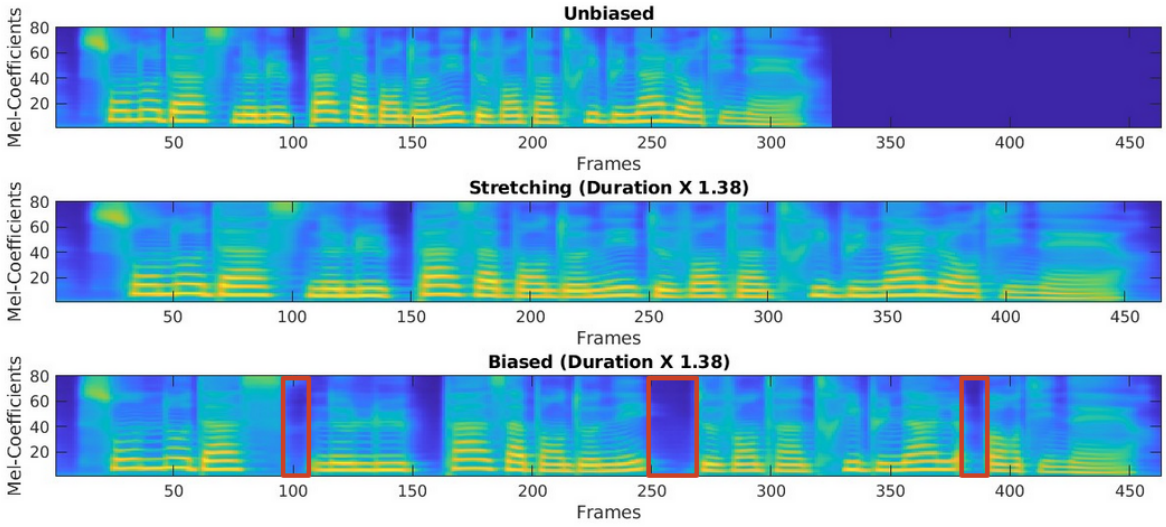


Figure 5.7: Illustration of the embedding bias duration control compared to the stretching for ***FS*** at the output of the text encoder. The utterance is: ",son audace ne l'eût pas abandonné devant un tribunal ordinaire;". Red squares indicate the addition of pauses with the embedding bias control.

5.4.2.1 Experimental Setup

To investigate the effect of the proposed embedding bias compared to explicit duration controls, we conducted a listening experiment where each model was evaluated against the ***Stretching*** method. Three controls are under study:

1. The duration control using **only the duration embedding bias**, corrected with a multiplying factor to match the target duration coefficient. These models are annotated $\boldsymbol{TC}^{\boldsymbol{dur}}$ and $\boldsymbol{FS}^{\boldsymbol{dur}}$ for Tacotron2 and FastSpeech2 respectively. $\boldsymbol{TC}^{\boldsymbol{dur}}$ and $\boldsymbol{FS}^{\boldsymbol{dur}}$ are the same models as $\boldsymbol{TC_B}$ and $\boldsymbol{FS_B}$ described in Section 5.2.3: the annotation $^{\mathbf{dur}}$ emphasizes that the control is only on the duration of phones.

2. The **combined control of duration and pauses**, without additional multiplying factors. The addition and deletion of pauses compensate for the actual elongation of phones not matching the target duration coefficient. These models are annotated $\boldsymbol{TC_P}^{\boldsymbol{dur\ +\ pause}}$ and $\boldsymbol{FS}^{\boldsymbol{dur\ +\ pause}}$ for Tacotron2 and FastSpeech2 respectively.

3. The explicit duration control of FastSpeech2, referred as $\boldsymbol{FS}^{\boldsymbol{explicit}}$.

A CMOS protocol was followed [International Telecommunication Union, ITU, 1998], where participants were presented with a pair <model|***Stretching*** > and asked which of these voice speed renderings felt the most natural. Scores are reported on a 7 discrete-point scale including three degrees of preference for each sound (1,2,3), and no preference (0). Each pair consisted of one sentence synthesized with one of the five models ($\boldsymbol{TC}^{\boldsymbol{dur}}$, $\boldsymbol{FS}^{\boldsymbol{dur}}$,

$\boldsymbol{TC_P}^{\boldsymbol{dur\ +\ pause}}$, $\boldsymbol{FS}^{\boldsymbol{dur\ +\ pause}}$ and $\boldsymbol{FS}^{\boldsymbol{explicit}}$) and one of the four duration coefficients (0.77, 0.87, 1.18, 1.44) against its ***Stretching*** counterpart. The duration control without pause bias was evaluated on its own during a previous experiment [Lenglet et al., 2022b], and is reported here as a comparison with the combined bias. $\boldsymbol{FS}^{\boldsymbol{explicit}}$ was therefore evaluated twice, the first time with $\boldsymbol{TC}^{\boldsymbol{dur}}$ and $\boldsymbol{FS}^{\boldsymbol{dur}}$, and then again with $\boldsymbol{TC_P}^{\boldsymbol{dur\ +\ pause}}$ and $\boldsymbol{FS}^{\boldsymbol{dur\ +\ pause}}$. The results from these two experiments are reported as $\boldsymbol{FS}^{\boldsymbol{explicit}\ 1}$ and $\boldsymbol{FS}^{\boldsymbol{explicit}\ 2}$, and serves as a reference to compare both evaluations. For fairness of comparison, all following results are reported by achieved elongation coefficient instead of target duration coefficient[7]. Order of presentation was randomly counterbalanced.

5.4.2.2 Results

83 participants[8] recruited on Prolific [Palan & Schitter, 2018] took part in the experiment, and each evaluated 72 stimuli following a Latin Square design so that every model, duration and sentence was equally heard by each subject. Fig. 5.8 reports the averaged CMOS obtained for each duration coefficient, by type of model and control. A positive value indicates that the model was preferred over ***Stretching***. A non-parametric Kruskal-Wallis test showed a significant effect of both duration control and models on the CMOS ($p < 0.01$). Post-hoc Wilcoxon tests by pairs were applied and a star (resp. 2 stars) in the figure indicates that the method shows a statistically different CMOS than the other method for this duration coefficient ($p < 0.05$ (resp. $p < 0.01$)).

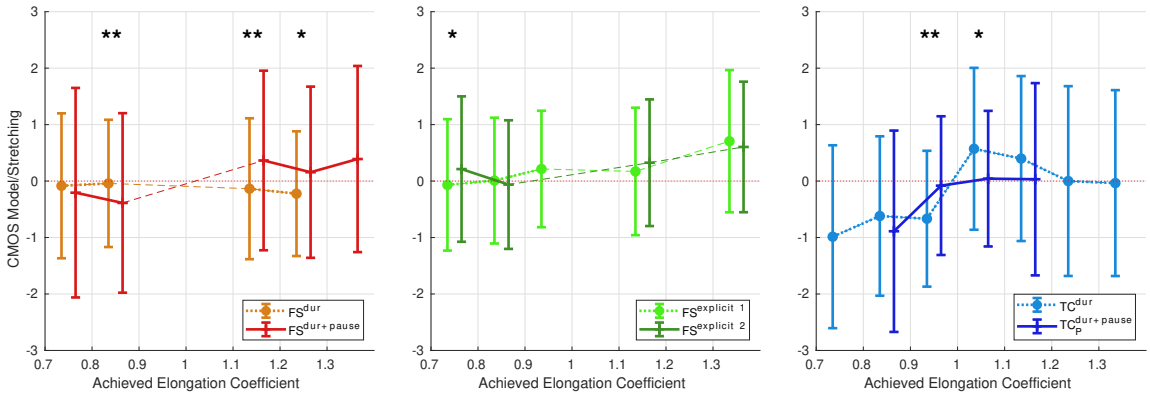


Figure 5.8: CMOS results of the evaluation of the systems without pause control (from Lenglet et al. [2022b], in orange light green and light blue) and with pause control (in red, dark green and dark blue).

[7] Achieved elongation coefficients are measured on the output signal as the ratio between the duration of the biased synthesis and the duration of the unbiased synthesis. In the figure, results are grouped by intervals of [0.1]

[8] 42 participants in the early experiment on $\boldsymbol{TC}^{\boldsymbol{dur}}$, $\boldsymbol{FS}^{\boldsymbol{dur}}$ and $\boldsymbol{FS}^{\boldsymbol{explicit}\ 1}$, and 41 later on $\boldsymbol{TC_P}^{\boldsymbol{dur\ +\ pause}}$, $\boldsymbol{FS}^{\boldsymbol{dur\ +\ pause}}$ and $\boldsymbol{FS}^{\boldsymbol{explicit}\ 2}$.

FS $^{explicit\ 1}$ and ***FS*** $^{explicit\ 2}$ exhibit similar results. The only statistical difference is evaluated for elongation coefficients in the range [0.7; 0.8], which may be explained by the selection of test utterances[9]. This confirms that the two experiments were performed with a similar reproducible setup. Thus, results of both experiments are compared in the following, and the explicit control is referred to as ***FS*** ^{explicit}.

Without pause embedding bias, ***FS*** ^{dur} was considered similar as ***Stretching*** while ***TC*** ^{dur} shows more contrasting results. For higher speaking rates, ***TC*** ^{dur} was significantly less preferred than ***Stretching***. A further analysis of the training set showed that highest speaking rates often correspond to the expressive reading of dialogs between characters. Without any residual encoder to segment this paralinguistic information apart from text input, ***TC*** ^{dur} may have learned an averaged representation of these characters, resulting in an unnatural speech depreciated by participants. By contrast, ***TC*** ^{dur} is preferred to ***Stretching*** on slightly lower speaking rates ([1; 1.2] of elongation). With this coefficient, the main difference between models lays in the non-linearity of phone duration (Fig. 5.5a), where ***TC*** ^{dur} closely matches the behavior of ***GT***. For very low speaking rates ([1.2; 1.4]), both embedding bias-controlled models are equally rated as ***Stretching***, while ***FS*** ^{explicit} is preferred.

We have emphasized the need for TTS models to introduce variations of phrasing when controlling the speaking rate. The results of ***FS*** $^{dur\ +\ pause}$ at low speaking rate validate our hypothesis: ***FS*** $^{dur\ +\ pause}$ is preferred to ***Stretching***. The limited control of pauses on Tacotron2 prevents the replication of theses findings with ***TC_P*** $^{dur\ +\ pause}$ because of the limited range of achieved elongation coefficients in this case. Conversely, higher speaking rates are rated lower than ***Stretching*** for ***FS*** $^{dur\ +\ pause}$. This indicates that the change of phrasing is perceived by participants, but is mostly depreciated.

Participants preferences were further explored by looking at the distribution of CMOS scores by model and by control methods. These distributions are illustrated with heatmaps in Fig 5.9. Scores distribution follows two distinct patterns. Participants' ratings on ***FS*** ^{dur} and ***FS*** ^{explicit} follow a normal distribution centered around 0. This means that most participants did not perceive any notable differences between the proposed control and the stretching. Conversely, scores of ***FS*** $^{dur\ +\ pause}$ and ***TC*** ^{dur} exhibit bi-modal distributions. This non-normality of distribution is confirmed with D'Agostino-Pearson's K2 test [D'Agostino & Pearson, 1973]. This signals that all utterances are not equally biased by the proposed method. We hypothesize that depending on the place of addition or deletion of a pause, perceived naturalness varies widely. Unfortunately, the proposed evaluation setup did not integrate any tools for participants to refine their evaluation locally. The difference was not significant for ***TC_P*** $^{dur\ +\ pause}$: only 4 out the 18 utterances evaluated achieved an elongation coefficients outside of the range [0.9: 1.1]. More utterances are needed to populate the two classes of examples observed.

[9] In case of significant increase of speaking rate, the explicit control may skip some phones which were originally short in the unbiased synthesis. This phenomenon, depreciated by participants, may have occurred more often in the evaluation of ***FS*** $^{explicit\ 1}$ than ***FS*** $^{explicit\ 2}$.

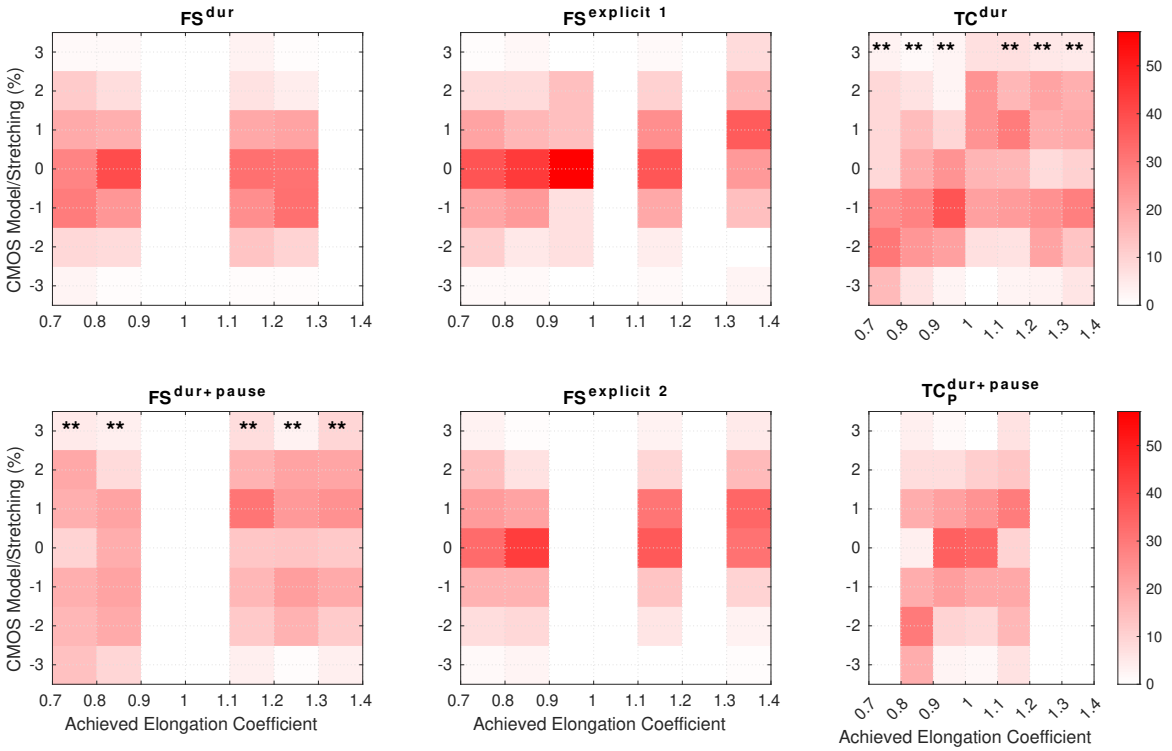


Figure 5.9: CMOS scores distribution. Positive scores indicate a preference for the control method compared to the stretching. ** indicates that the distribution for this elongation coefficient is not normal ($p<0.01$).

5.4.3 Discussion: More natural speaking rate control with combined embedding biases.

Pause addition and deletion is an important part of speaking rate control in natural speech, which is mostly ignored by neural TTS control. The proposed method enables us to modify pause proportion in synthesis along with a more natural elongation of phone duration. Perceptual evaluations partially validated the benefits of this control, but also highlighted the need for clearer constraints on where in the utterance to authorize pauses additions and/or deletions. Part Of Speech (POS) tags could be implemented to limit the unnatural renditions of pauses.

Additionally, the lack of insight into perceptual scores given by participants in the evaluation reinforced our need for more local tools to evaluate synthetic speech. For instance, our evaluation would have benefited from the use of Rapid Prosody Transcription (RPT) [Cole & Shattuck-Hufnagel, 2016] as a way for participants to indicate the parts of the utterance they based their score on. Interfaces have been developed to collect these data [Gutierrez et al., 2021], which we could consider to enrich our evaluation setup.

5.5 Discussion: Cost Efficient Control by Linear Biases

The linear tracking method proposed in Chapter 4 revealed that acoustic and prosodic features were partially linearly encoded in latent embeddings of neural TTS models. This analysis highlighted the main directions in embeddings which encode the variations of the selected set of local acoustic features. These directions were used in this chapter in order to introduce linear biases, called **Embedding Biases**, to gain explicit control over the measured acoustic features.

5.5.1 Multi-Level Feature Encoding in Embeddings

The evaluation of the proposed control showed that all measured acoustic features are controllable to a certain extent in at least one layer of the architecture. However, the control range varies widely between features. We have shown that goodness of fit is a necessary but not sufficient condition for controllability. From Fig. 4.5a and Fig. 5.2d, energy and pitch seem to be linearly encoded from the output of the encoder to the last layer of the decoder for $\boldsymbol{FS}$ and $\boldsymbol{FS}_{\setminus \boldsymbol{E}}$. However the control through linear bias is only possible in the last layer of the decoder for energy in $\boldsymbol{FS}_{\setminus \boldsymbol{E}}$, and does not produce any effect for pitch in the decoder of $\boldsymbol{FS}$. Linear representations being encoded do not mean that the model uses these representation as is. Several representations of the same feature may coexist in embeddings. This was also observed in VAE-latent spaces, with the co-encoding of between-mode and within-mode variability of F0 representations [Jacquelin et al., 2023].

This is particularly visible in $\boldsymbol{FS}$, in which the pitch embeddings summed to the output of the encoder do not erase the linear pitch representations from the text embeddings, but rather add another level of pitch encoding. This multi-level encoding may help the model avoiding feature loss from one layer to another, which could be caused by the non-linearity of multi-head Transformer blocks, or may be a side effect of the forced-dropout during the training [Hinton et al., 2012]. The challenge of the linear tracking is then to carefully target the dimensions that encode the feature to modify. The refined linear predictor training described in Section 4.1.2 already improved the selection of dimensions compared to the straightforward method used by Lenglet et al. [2022b]. Nonetheless, future works should further improve the targeting of dimensions of interest in order to provide a better control of acoustic features, both to grant wider ranges of control and to limit the unwanted impact on other encoded representations.

5.5.2 Universal and Cost-Efficient Control Mechanism

Despite the limits of the proposed control, the universality of the method makes it attractive to design cost-efficient control mechanisms for any continuous acoustic features. The proposed control does not require any additional training of the neural models, nor additional data. This control was illustrated for Tacotron2 and FastSpeech2, but could be applied to any

encoder-decoder TTS architecture. Additionally, as illustrated in Section 5.3, the proposed control takes advantage of covariations and non-linearities statistically learned by neural models during the training phase. As such embedding biases provide more natural control over disentangled explicit methods.

Furthermore, the proposed control can be adapted to discrete phonological features, as illustrated in Section 5.4 for pause control. The control of liaisons has also been tested with the proposed approach. The liaison embedding bias can also be used to modify the proportion of liaisons produced by the models. However, this control was not further evaluated.

5.5.3 Adaptation to Style Control

This chapter has illustrated how the better understanding of latent representations computed by neural models opens the route toward the design of new careful control mechanisms for TTS. This analysis was applied at the text embedding-level, considered as the atomic building block of speech representations computed by TTS models. However, the proposed analysis methods are not restricted to this use-case. Utterance-wise style or speaker biases may be subjected to similar analysis, provided that acoustic features are adapted to match the utterance-scale. The adaptation of the proposed method to utterance-wise biases could help exploring the utterance-embedding space in order to extrapolate control out of the limited number of examples seen during training.

Finally, the proposed control method emphasizes how a linear bias may have a non-linear impact on the generative process. This non-linearity was illustrated on pitch control on Fig 5.3, and further evaluated on duration control in Section 5.3.2 as an extension of our paper [Lenglet et al., 2022b]. As a matter of fact, addition of utterance-style biases has been widely used in the expressive control literature [Y. Wang et al., 2018; Wu et al., 2019; Y.-J. Zhang et al., 2019], showing more complex acoustic modifications applied than uniform offsets along the utterance. Moreover, the better supra-segmental control compared to segmental advocates for prosodic cues being encoded in shared directions across phone-classes. In that regard, these observations support the potential of utterance-wise style bias to modulate fine-grained prosody. However, such utterance-wise style biased disentangled from the text input may lack the support of the syntactic and semantic content to determine the prosodic targets to apply. While these targets may be predicted by human expertise and applied via explicit mechanisms such as the control method proposed in this chapter, most real-life applications cannot afford the prediction of prosodic contours through human-in-the-loop. Therefore, we will also explore implicit expressive bias approaches in the following chapter.

Chapter 6

Expressive Control from Local Speech Units

Contents

Chapter Highlights

This chapter presents our proposed **Local Style Tokens (LST)** layer to modulate utterance-wise attitude embeddings from the semantic and syntactic structure of the textual input. The proposed LST bridges the gap between linguistic and paralinguistic prosodic modeling. The LST is integrated in a GST-enhanced FastSpeech2 with constrained tokens. We show that this auxiliary layer introduces **local prosodic modulations**. Perceptual evaluations confirm the benefit of the LST over a GST baseline, both with **word-level and phone-level** modeling.

Related contributions: [Bailly et al., 2024; Lenglet et al., 2023b]

This sixth chapter describes our contribution to better model local modulations of prosody for expressive speech synthesis. As elaborated in Section 1.2.2, expressive speech synthesis is mostly addressed in the literature by the combination of the embeddings outputted by the text-encoder with an utterance-wise bias [Skerry-Ryan et al., 2018; Y. Wang et al., 2018; Y.-J. Zhang et al., 2019]. This utterance-wise bias, generally referred to as a **style embedding**, is supposed to capture the speech variability otherwise unexplained by the text input or the speaker identity, namely the **prosody**. However, natural expressive speech relies on multiple levels of variations. As a result, utterance-wise style embeddings may lack finer-grained representations in order to fully mimic natural voice behavior.

Ground-breaking performance of models like Global Style Tokens (GST) [Y. Wang et al., 2018] advocate for utterance-wise style embeddings' ability to reproduce natural complex prosodic patterns beyond simple offsets of acoustic parameters. In Chapters 4 and 5, we have demonstrated that internal embedding biases computed from model latent spaces were also able to control the prosodic features produced by TTS models. Our evaluation highlighted the non-linear effects produced by the addition of such linear biases. These findings support the hypothesis that non-linear operations computed into neural layers are able to turn a linear bias into adequate local modulations of synthetic speech.

However, most style control architectures (GST included) assume that style contributions are disentangled from the input text. This assumption makes sense in TTS use cases: the aim of such models is to produce any text with any style. In order to achieve this disentanglement, neural architectures are designed to minimize the linguistic information leakage from reference audio samples to prosodic representations. Yet, the natural prosodic structure of one's speech not only depends on one's intents or style, but also on the content itself, as syntactic and semantic structures play an important role in the organization of stress and phrasing [Liberman & Prince, 1977; Selkirk, 1986]. That is why we believe that stylistic embeddings could benefit from relying on the text itself. Thus, this chapter describes our proposition of an auxiliary neural module called **Local Style Tokens (LST)** to tailor stylistic representations to the textual input content.

We evaluated the benefits of the proposed LST module compared to the GST architecture in Lenglet et al. [2023b]. All expressive TTS variants were implemented on top of our FastSpeech2 baseline established in Chapter 2. The proposed LST module could be applied to any TTS architectures, but the choice of FastSpeech2, as well as the main design choices are further discussed in Section 6.1. The rest of this chapter presents the details of this module and an extended version of our results. Section 6.2 describes the expressive corpus we recorded to train our expressive TTS. The implementation of the LST module is further explained in Section 6.3, followed by our complete evaluation in Section 6.4. Our perspectives of extending this control to a wider range of explicit labels is discussed in Section 6.5.

6.1 Design Choices of the LST Module

6.1.1 From Embedding Bias to Attitude Control

The acoustic control provided by the embedding bias in Chapter 5 is limited for general expressive control of propositional attitudes. Since the embedding bias method acts on mean variations of acoustic features, target variations are required beforehand in order for a human operator to implement the correspondence between expected styles and acoustic biases. Such correspondence is inherently model-dependent, which cannot scale to continuous training workflows[1]. This is the reason why we decided to integrate the constrained-GST module [Wu et al., 2019] as a way to learn unsupervised utterance-wise acoustic and prosodic representations. Nonetheless, embedding biases have provided interesting insights into the encoding of segmental and supra-segmental features that ultimately oriented the design of the our proposed LST module:

1. FastSpeech2 was chosen as the baseline architecture for the implementation of the expressive control. Although the implementation of prosodic predictors does not provide a better control of pitch and energy for FastSpeech2 than Tacotron2 (see Fig. 5.2c and 5.2d), the robustness of the duration predictor coupled with the length regulator compared to the attention mechanism of Tacotron2 provides a much better control of the phrasing (see Fig. 5.2e and 5.2f). Non-attentive Tacotron2 [Shen et al., 2020] should be considered as an alternative for future work.

2. The impact of the phonetic prediction layer on phrasing modulations by embedding bias validated the implementation of this sub-task for expressive synthesis. Phrasing plays an important role in the expressiveness of speech [Godde et al., 2017] and is better modeled by $\boldsymbol{FS}$ than $\boldsymbol{FS}_{\backslash \boldsymbol{phon}}$ (see Fig. 5.2f).

3. The tracking of features by layer indicated that the output of the encoder provides a common encoding space for prosodic representations. Thus, the output of the encoder appears to be the best layer to implement the LST module.

4. As seen in Section 5.2.2, segmental features are less controllable by linear biases, because the direction of encoding of segmental features may vary depending on the phone classes. The introduction of phone-level style embeddings should compensate for this effect.

To be more specific, the LST module extends the GST implementation to the segmental level. Through the unsupervised training of local prosodic embeddings, the LST module aims at learning finer-grained prosodic patterns based on shorter speech units, like words or phones. On the one hand, word-level units enable prosodic embeddings to rely on the syntactic and semantic structure of the text content, provided that the text encoder is able to learn and encode this information in individual character or phone embeddings. This hypothesis is supported by the disambiguation of homographs achieved at the output of the text encoder. The performances reported in Section 2.4.3 advocate for the ability of the FastSpeech2 encoder to distinguish homographs beyond the part-of-speech properties, which could be assimilated

[1] Models used for deployed applications are generally trained/fine-tuned on new data on a regular basis.

to partial semantic representations. On the other hand, the phone-level is expected to ease the control of segmental acoustic features, but also introduces direct manipulations of phonetic classes. The replacement of vowels by schwas, vowel harmonic choices, liaisons and pauses are all dictated by phonetic representations encoded at the output of the text encoder. Phone-level style control enables direct manipulation of these representations without affecting neighbor embeddings.

6.1.2 Local Prosodic Modulations in the Literature

Fine-grained prosodic representations have been proposed for TTS before. By construction, the pitch and energy embeddings in FastSpeech2 variance adaptors [Ren et al., 2021] are spectrogram frame-level prosodic embeddings. These provide some prosodic control during inference, but also help better modeling of fundamental frequency. The LST module relies on the same mechanism as the prosodic predictors, by re-injecting prosodic representations within the model. The introduction of such local prosodic representations has already been confirmed to benefit to prosodic transfer [L.-W. Chen & Rudnicky, 2022], but those authors did specifically try to disentangle local prosodic contributions from the textual content, which contrasts with our goal to model the interaction between propositional attitudes and the syntactic and semantic structure of the utterance.

More focused toward expressive control, as described in Section 1.2, Klapsas et al. [2021] proposed to enhance Tacotron2 [Shen et al., 2018] with word-level style embeddings that are concatenated to the encoder output. Word-level representations are computed with recurrent layers, and then passed to a style attention layer similar to GST [Y. Wang et al., 2018]. This work inspired us for the present study, but we tried to avoid its main limitation: the authors had to train a Prior Encoder, which predicts word style embeddings from the text input in order to synthesize text without an audio reference. As a result, the output synthesis is solely based on the text input, denying the choice of expressive style during inference. In that regard, this approach is similar to the Local Style Tokens computed by Veaux et al. [2023]. They trained local tokens on word-level representations computed from Continuous Wavelet Transforms [Vainio et al., 2013]. During inference, local token attention weights are inferred from pre-trained BERT representations [Devlin et al., 2018]. This approach extends the prediction of global style token attention weights from Stanton et al. [2018]. The methods described in these studies are constrained to local prosodic representations derived from the text, which limits their capabilities to the prediction of linguistic prosody. On the contrary, our work aims at bridging the gap between linguistic and paralinguistic prosody, by the introduction of the LST module as a way to modulate paralinguistic contributions based on the text structure and the attitude to produce.

Hierarchical TTS models like CHiVE [Kenter et al., 2019] or MsEmoTTS [Y. Lei et al., 2022] also take advantage of the multi-level aspect of speech, by combining intermediate representations from different scales: phones, syllables, words, utterance, etc. The entire architectures of these models are built on this hierarchical representation. On the other hand, the proposed LST module is independent; it can be plugged into any encoder-decoder TTS architecture, also with various scopes of representation.

6.2 Dataset Description

As a first step in the adaptation of our baseline French TTS described in Chapter 2 to expressive synthesis, we needed an expressive-labeled French dataset. Att-Hack [Le Moine & Obin, 2020] has been proposed as an expressive dataset designed to study the inter-speaker acoustic variability of speech attitudes production. This dataset provides recordings for 4 social attitudes (friendly, seductive, dominant, distant) uttered by 25 speakers in 100 utterances. This corpus provides 3 to 5 variations by speaker/utterance/attitudes. This variability of production is an asset for prosodic analysis. However, 100 utterances are not enough to provide a extensive phonetic coverage, and this limited number of utterances by speaker reduces the potential of this corpus for TTS training. Additionally, social attitudes define the relationship between the speaker and his/her interlocutor.

As described in Section 1.2, the current focus of the presented work is instead propositional attitude, which defines the attitude of the speaker towards the utterance he/she is producing. As such, the selection of attitudes of the Att-Hack corpus did not fit our need. Similarly, in the SynPaFlex corpus [Sini et al., 2018], the expressive labels from the "Basic Emotions" [Ekman et al., 1999] describe the mental state of the speaker, which we distinguish from propositional attitude. In absence of a suitable French corpus, our own expressive dataset was recorded by Gérard Bailly[2] and Frédéric Elisei. The recording process is described in Appendix A.2.1. Table 6.1 summarizes the train/test set distribution by attitude used in this section.

Table 6.1: Duration and segmentation of our single-speaker expressive Dataset (see Appendix A.2.1 for further details). Durations are given in hh:mm:ss.

Style		**Train**		**Test**	
English	**French**	**Duration**	**# Utt**	**Duration**	**# Utt**
Angry	Colère	00:24:12	523	00:01:30	32
Comforting	Réconfortant·e	00:32:18	488	00:01:36	27
Committed	Déterminé·e	00:21:06	430	00:01:24	29
Enthusiastic	Enthousiaste	00:29:30	569	00:01:24	28
Obvious	Evidence	00:27:00	492	00:01:30	27
Playful	Espiègle	00:19:06	465	00:01:30	28
Pleading	Suppliant·e	00:34:12	605	00:01:54	31
Skeptical	Incrédule	00:29:48	620	00:01:36	32
Sorry	Désolé·e	00:24:12	448	00:01:06	23
Surprised	Surpris·e	00:26:48	503	00:01:36	32
Thoughtful	Pensif·ive	00:43:24	450	00:02:06	27
Narrative	Narratif	04:47:36	6235	00:14:36	307
	Total	**09:59:48**	**11828**	**00:31:48**	**633**

[2]PhD director

6.3 Integration of the LST Module

This section describes the architecture of the proposed Local Style Tokens (LST) module and how it is integrated in the GST-enhanced FastSpeech2 pipeline. The overall architecture of the proposed model is shown in Fig. 6.1.

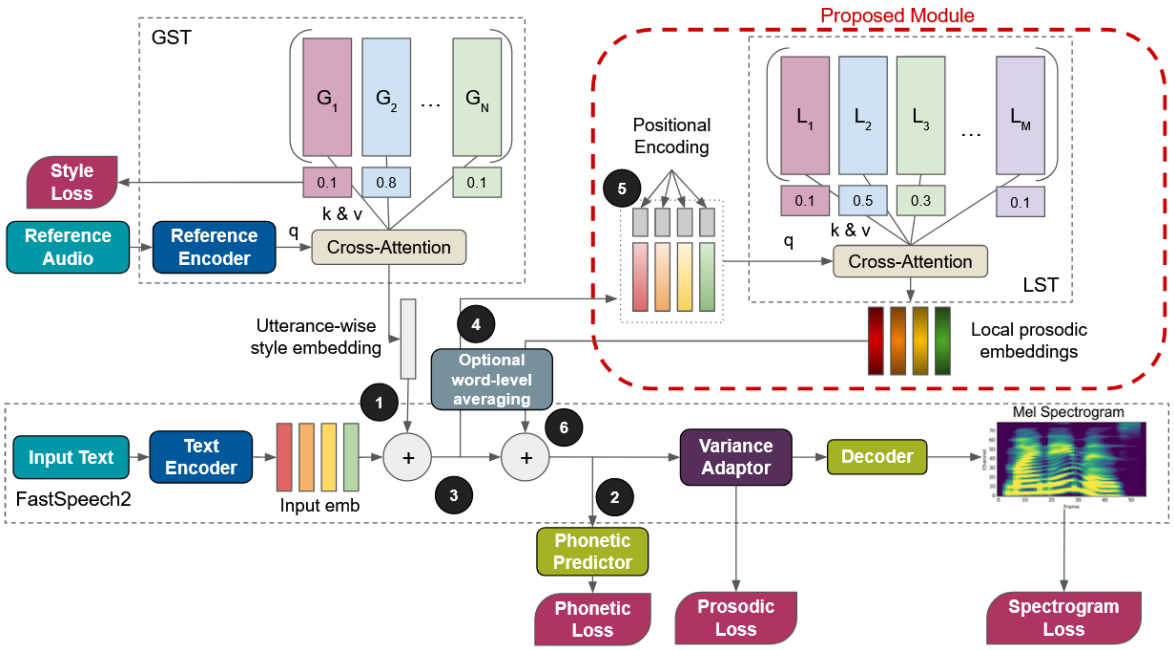


Figure 6.1: Integration of the Local Style Tokens Module in the FastSpeech2-GST architecture.

6.3.1 Model Backbone

The backbone of the model is FastSpeech2 [Ren et al., 2021], whose encoder, variance adaptor and decoder are kept unchanged[3]. In addition, a label-constrained GST module [Wu et al., 2019] is implemented at the output of the text encoder (Fig. 6.1, label 1)[4]. This constrained-GST module converts a reference audio sample into a fixed-size vector through a reference encoder [Skerry-Ryan et al., 2018]. This fixed-size vector is then used as the query of the cross-attention mechanism in an attitude token layer. Similarly to GST [Y. Wang et al., 2018], this attitude token layer computes weights that measure the similarity between the reference vector and each global style token. Following Wu et al. [2019], a cross-entropy loss is added to enforce each token to encode one particular attitude. The weighted sum of tokens is then added to all phone embeddings computed by the text encoder.

Contrary to a set of learnable style embeddings, this module helps training the model on heterogeneous style samples. When given the same style as target, one speaker may produce highly variable utterances, with varying degrees of the given style. The constrained-GST module may account for this degree by using a mixture of tokens for low degree utterances,

[3]https://github.com/ming024/FastSpeech2
[4]GST implementation based on https://github.com/taneliang/gst-tacotron2

even though their label is the same as unambiguous utterances. This layer is expected to mitigated the effect of mislabelled utterances, as described in Section 6.5.

Following the FastSpeech2 baseline established in Chapter 2, this model is trained on mixed inputs. The phonetic prediction layer is implemented after the contribution of the LST module (Fig. 6.1, label 2), since the LST is expected to impact phonological behavior.

6.3.2 Local Style Tokens Module

The Local Style Tokens architecture (LST) is introduced as an auxiliary module which further modulates the output of the text encoder. Although this module does not need to be combined with the GST module, the LST module alone does not provide explicit control of the synthesis style at inference, which is why the constrained-GST is kept in this model. The LST module may be seen as a residual layer which modulates the latent representations that have been uniformly biased by the GST embedding, according to the content or the position of linguistic units in the utterance. The LST modulation further improves acoustic and prosodic representations in this latent space.

The LST module follows the same architecture as the original GST [Y. Wang et al., 2018]. Two levels of local tokens are examined in this study: Word-level and Phone-Level. In the case of Phone-level tokens, this module takes as inputs the globally-biased phone embedding sequence (Fig. 6.1, label 3). For Word-level, this sequence is averaged by word, to compute word-level representations (Fig. 6.1, label 4). Because our dataset preserves word boundaries and punctuation marks in case of phonetic inputs, pseudo-word representations are also computed for spaces and punctuation marks (or both when consecutive), also by averaging embeddings. Note that local biases added to word boundaries provide an unsupervised alternative to the pause embedding bias evaluated in Section 5.4.

Acoustic patterns relative to style generation depend on the syntactic structure of the utterance and on the relative position of units in the utterance. However, similarly to GST [Y. Wang et al., 2018], the cross-attention mechanism in the LST module uses dot product attention, which cannot infer relative positions of representations in the input sequence, unlike recurrent networks. Also, we demonstrated that the positional information brought by the positional encoding that is added to phone embeddings in the text encoder quickly fades in the successive layers of the FastSpeech2 encoder as illustrated in Fig. 4.5a. Thus, a 32-dimensional positional encoding is re-introduced with concatenation at the input of the LST module (Fig. 6.1, label 5).

The resulting input tensor serves as a set of queries for the cross-attention mechanism in the LST module. A set of weights is computed for each element in the sequence, and the time-dependent weighted sum of token values constitutes the local prosodic embedding sequence which is summed to the globally-biased phone embedding before the variance adaptor (Fig. 6.1, label 6). In case of Word-level LST, each local prosodic embedding vector is first duplicated to be added to all phones in the given word (resp. pseudo-word). For ease of interpretability of local token weights, the cross-attention mechanism is single-headed.

6.3.3 Training and Inference Processes

During training, the reference mel-spectrogram is identical to the target output. The reference encoder and the cross-attention GST work as an attitude recognition module which computes a probability distribution on all constrained style tokens from the given audio input. In contrast, the LST weights are unsupervised during training, i.e., the LST module does not require any additional loss. It is trained by the back-propagation of the spectrogram loss, prosodic predictors losses and phonetic loss. The back-propagation is not stopped at the input of the LST module, which enables the text encoder to incorporate features that may be used to compute local prosodic embeddings in the LST module. The entire model can be trained jointly, from scratch.

Similarly to constrained-GST, two style control methods are available during inference: 1) use of a target reference audio which produces a mixture of global style tokens or 2) specify the mixture of global style tokens to use. Because the GST module is constrained, each global style token has been trained to produce one particular style. Thus, one-hot vectors are particularly fitted to generate the desired style. Nonetheless, a mixture of global tokens can also be used to provide less caricatural biases. Local prosodic embeddings are computed in parallel by the LST module, which does not impact the inference speed of the model.

6.4 Evaluation of the LST Module

Three models were trained for this study: 1) FastSpeech2 with constrained-GST referred to as $\boldsymbol{GST}$ (the Baseline) ; 2) The Baseline enhanced with word-level LST referred as $\boldsymbol{LST_W}$; and 3) The Baseline enhanced with phone-level LST referred as $\boldsymbol{LST_P}$. All models were trained on the same segmentation of train/test set, whose details are given in Table 6.1. Following the constrained-GST architecture given by Wu et al. [2019], 12 tokens are needed in the $\boldsymbol{GST}$ layer to account for each style label of Table 6.1. The target styles given to the actress are used as style labels. The training follows the procedure given in Appendix C. The vocoder used is Waveglow, as described in Appendix D.2.

6.4.1 Local Token Usage

Fig. 6.2 illustrates an example of the local contribution of LST. The utterance "§Cependant, les critères fixés pour l'attribution des subventions sont #très# restrictifs.§" is taken as example. Pitch contours predicted by the pitch predictor for $\boldsymbol{LST_W}$ and $\boldsymbol{GST}$ are displayed for the style “Committed”. Note that the LST module uses the text embeddings biased by the GST module as input, so pitch contours mostly follow the same pattern. However, subtle differences appear: $\boldsymbol{LST_W}$ provides a better balance between accentuated words like "fixés" and the rest of the utterance. Emphasis, indicated by symbols "#", is produced by both models, but $\boldsymbol{LST_W}$ shows a more consistent pitch on the predicate adjective group "très restrictifs". Future studies should focus on the identification of acoustic features into local prosodic embeddings

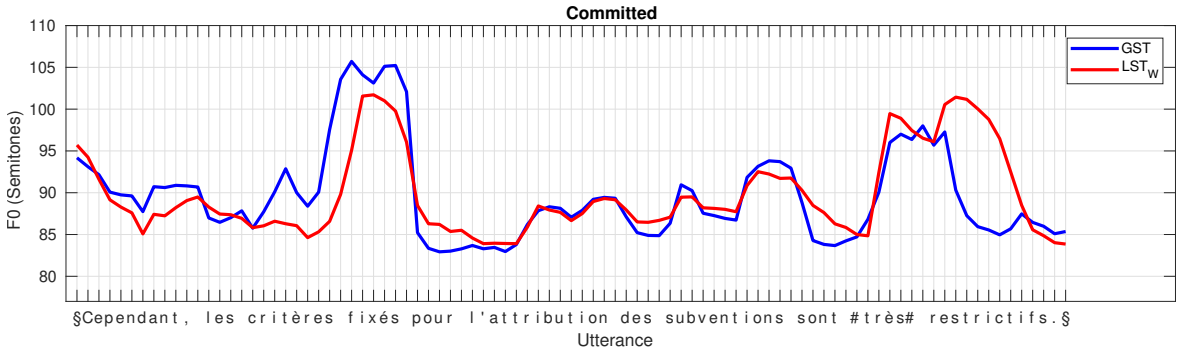


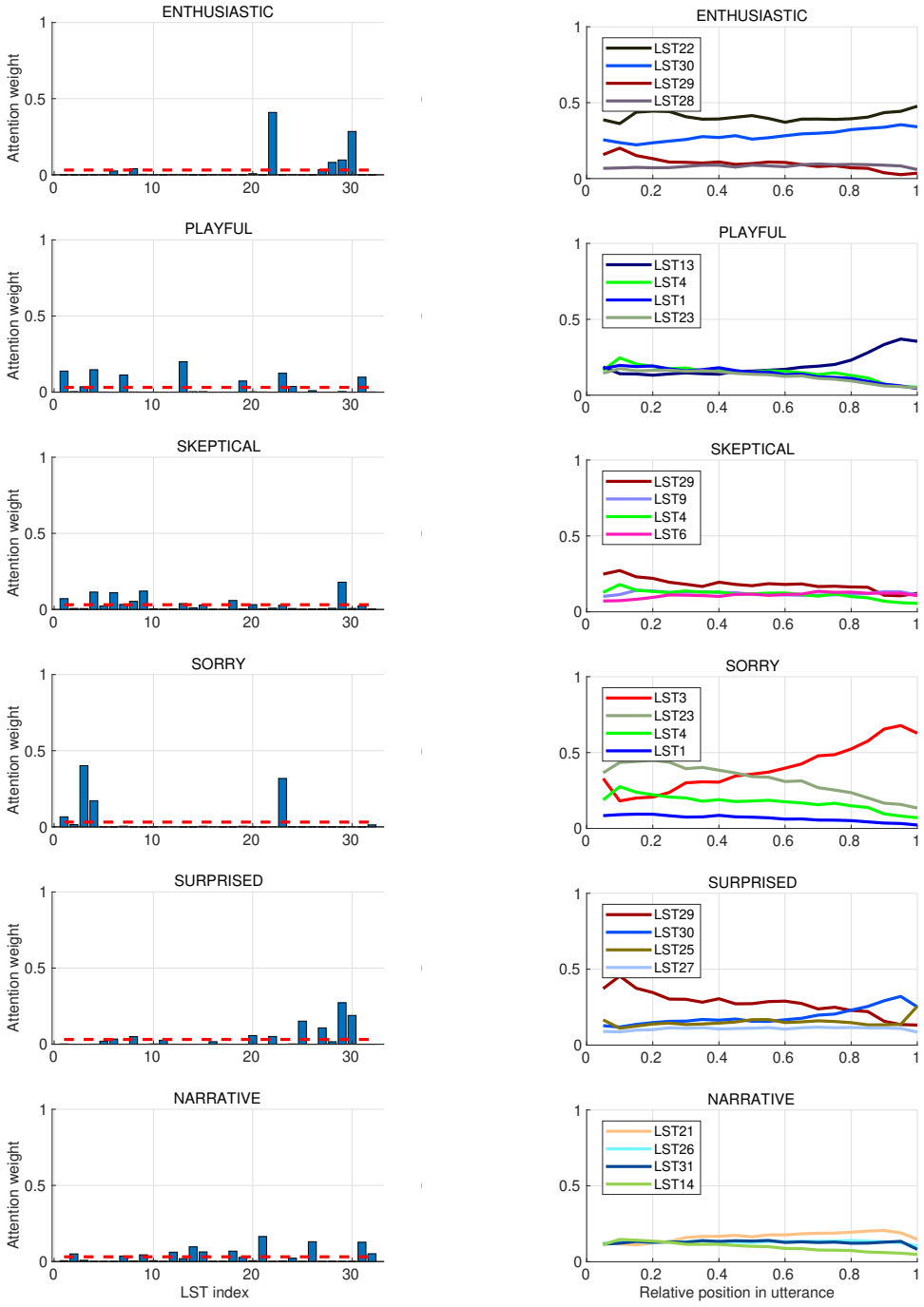
Figure 6.2: Comparison of pitch contours predicted by $\boldsymbol{LST_W}$ and $\boldsymbol{GST}$. "Committed" is illustrated as an example of style.

learned by the LST module. The analysis methods developed in Chapter 4 could similarly be applied to local prosodic embeddings, but were left for future work due to lack of time.

The number of local style tokens is fixed to 32 for both $\boldsymbol{LST_W}$ and $\boldsymbol{LST_P}$. 32 tokens is chosen as a middle ground between sharing tokens across global representations and providing enough local tokens so that each global style can rely on dedicated local tokens. A related implementation of local tokens [Veaux et al., 2023] also used 32 tokens.

To evaluate the usage of each individual local token, 100 utterances of the test set were randomly selected, and each utterance was generated with the 12 styles of the corpus. Mean attention weights of each local token were computed per style. Examples of mean attention weights by local token and the dynamic of this attention are given in Fig. 6.3 for $\boldsymbol{LST_W}$. Table 6.2 summarizes the number of local tokens used per style, as well as the number of tokens that are exclusive to the specified style. One local token is counted as used if its mean activation weight is above the uniform distribution across all local tokens (above the red dashed line in Fig. 6.3a). The overall number of tokens used differs from the sum because some tokens are shared across styles. Two tokens are never used by $\boldsymbol{LST_P}$. $\boldsymbol{LST_W}$ and $\boldsymbol{LST_P}$ with 64 tokens were tested but showed that too many tokens were never used.

The diversity of local token usage illustrates the benefits of modeling prosody at a smaller scale. Multiple local tokens are used by all styles to model various local patterns. "Angry", "Comforting", "Playful" and "Narrative" use exclusive local tokens in $\boldsymbol{LST_W}$, assessing for unique speech behaviors in this sub-corpus (same for "Narrative" in $\boldsymbol{LST_P}$). Fig. 6.3b shows the dynamic of local token attention relative to the position in the utterance. Global styles exhibit various patterns, but most characteristic behaviors are found at the beginning (LST29 for "Surprised") and at the end of utterances (LST13 for "Playful" and LST3 for "Sorry"). Other styles like "Skeptical" are more stable, but smaller variations of local token usage also indicate that the LST module helps with modulating representations at a finer grain.



(a) Mean LST usage by style.

(b) Mean LST usage relative to the position in utterance (0: first character, 1: last character).

Figure 6.3: Local Style Tokens usage by style for $\boldsymbol{LST_W}$. Six styles are shown as examples: "Enthusiastic", "Playful", "Skeptical", "Sorry", "Surprised" and "Narrative". Only the contours of the 4 local tokens with the maximum mean attention weights are shown.

Table 6.2: Number of Local Style Tokens used by the model per style.

Style	LST_W		LST_P	
	# Tokens	# Exclusive	# Tokens	# Exclusive
Angry	9	1	8	0
Comforting	8	1	7	0
Committed	8	0	11	0
Enthusiastic	6	0	10	0
Obvious	7	0	8	0
Playful	9	1	11	0
Pleading	6	0	10	0
Skeptical	10	0	12	0
Sorry	4	0	8	0
Surprised	8	0	11	0
Thoughtful	8	0	12	0
Narrative	11	3	13	2
Overall (/32)	**32**	**6**	**30**	**2**

6.4.2 Objective Evaluation

Objective evaluations of the synthetic models were conducted to assess the benefits of the proposed model compared to the baseline. Models were evaluated on three aspects: training loss criteria, pitch variations and phrasing behaviors. All statistical differences between distributions are evaluated pair-wise through non-parametric Wilcoxon rank sum tests. The objective metrics shown in this section focus on various evaluations of the three main prosodic features: duration, pitch and energy. Other acoustic features like voice quality may impact style modeling [Gobl et al., 2002], but were not measured in this study.

6.4.2.1 Test Set Errors

All models are trained under the same loss criteria, which include mel-spectrogram losses and prosodic features predictions (duration, pitch and energy). Table 6.3 summarizes the errors calculated on the test set ground truth ($\boldsymbol{GT}$) after training. Spectral error is computed on synthesis aligned with $\boldsymbol{GT}$ using Dynamic Time Warping (DTW) [Kubichek, 1993]. Mean Euclidean distances are evaluated on the alignment path. Duration and energy errors are computed on all phones, while pitch error is only evaluated on vowels.

Lower errors indicate that models that implement the LST modules produce speech closer to the $\boldsymbol{GT}$ for most styles. Over all errors, $\boldsymbol{LST_W}$ provides the most consistent benefits, with 28 improvements and 14 degradations, compared to 20 improvements and 20 degradations for $\boldsymbol{LST_P}$. These improvements were significant for "Narrative", but not for the other styles. "Committed", "Enthusiastic", "Pleading", "Surprised" and "Narrative" are the most improved styles. This indicates that those five styles rely on local prosodic patterns that are difficult to

Table 6.3: Mean errors per style computed on the test set. Blue (resp. red) indicates a lower error (resp. higher error) than ***GST***. * and ** indicate that the distribution statistically differs from the ***GST*** baseline with p<0.05 and p<0.01, respectively.

Style	Spectral Error (dB)			Duration Error (ms)			Pitch Error (Semitones)			Energy Error (dB)		
	GST	***LST$_W$***	***LST$_P$***	***GST***	***LST$_W$***	***LST$_P$***	***GST***	***LST$_W$***	***LST$_P$***	***GST***	***LST$_W$***	***LST$_P$***
Angry	0.93	0.91	0.93	9.18	9.01	9.13	2.29	2.14	2.50	3.24	3.15	3.34
Comforting	0.88	0.88	0.89	11.41	11.64	11.12	1.59	1.62	1.77	3.12	3.24	3.11
Committed	1.00	0.99	0.96	9.83	9.83	9.33	4.10	4.15	3.93	3.26	3.16	3.24
Enthusiastic	1.18	1.16	1.18	9.82	9.45	9.82	4.31	3.93	4.31	3.03	3.10	3.06
Obvious	1.07	1.05	1.08	10.74	10.23	10.01	3.32	3.12	3.41	3.12	3.09	3.12
Playful	0.97	0.99	0.96	12.08	11.29	11.25	4.06	3.98	4.11	3.09	3.06	3.04
Pleading	0.92	0.92	0.91	9.67	9.03	9.49	1.93	1.78	1.72	2.49	2.44	2.45
Skeptical	0.98	0.97	0.99	10.10	9.85	10.13	2.86	3.03	3.08	3.06	3.03	**3.23****
Sorry	0.67	0.68	0.67	9.50	9.50	9.70	1.05	1.16	1.04	2.64	2.75	2.77
Surprised	0.97	0.97	0.97	10.20	9.85	10.07	3.47	3.33	3.40	3.21	3.13	3.30
Thoughtful	0.94	0.95	0.97	22.23	22.28	22.52	2.51	2.63	2.52	2.86	2.91	2.96
Narrative	0.90	0.90	0.89	10.45	**10.36***	10.53	2.75	2.73	2.74	2.93	**2.86***	**2.86****
Total	0.93	0.93	0.92	10.52	10.31	10.44	2.70	2.67	2.71	2.97	2.94	2.96

model with utterance-wise style representation. On the other hand, "Comforting", "Skeptical", "Sorry" and "Thoughtful" show higher errors with LST. Overall, the worse results of ***LST$_P$*** may be explained by the wider variability provided by local tokens at the phone scale. This variability opens the door for more risks of divergence with ***GT***.

While lower errors indicate that synthetic speech is closer to the natural utterances recorded in our corpus, there is no golden standard for conveying a given style. Many variants: 1) could have been performed by the recorded speaker for this same sentence and style, and 2) may be perceived as similarly expressive for a human listener. As a result, the ***GT*** is not the only correct speech production, and alternative objective evaluations are needed to assess the expressive quality of the synthetic speech. In the following, we then compare distributions of prosodic parameters measured on ***GT*** and on each of our models. Our criterion for a successful rendering of prosodic features is therefore to have **non**-significant differences between a model and the ***GT***.

6.4.2.2 Pitch standard deviation

Pitch standard deviation by utterance is commonly used to evaluated expressive capabilities of TTS models [Klapsas et al., 2021; Ren et al., 2021]. Table 6.4 compares the pitch variability of ***GT*** to that of the synthetic models. Highly variable styles like "Enthusiastic", "Obvious", "Playful", "Skeptical" and "Surprised" are harder to model for TTS, as shown by statistical differences between ***GT*** and all synthetic models. Overall, the LST module leads to more pitch variability, even though results were not significant. Significant improvements were found for "Sorry", with ***LST$_W$*** generating pitch standard deviations closer to ***GT***.

Table 6.4: Mean standard deviation of pitch per style (Semitones). * indicates that the distribution statistically differs from the ***GT*** ($p < 0.05$). Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the ***GST*** baseline.

Style	***GT***	***GST***	***LST$_W$***	***LST$_P$***
Angry	3.59	**2.95**	**2.85***	**2.77***
Comforting	1.94	1.68	1.70	1.77
Committed	3.92	3.92	3.94	3.69
Enthusiastic	4.79	**3.27***	**3.44***	**3.24***
Obvious	4.25	**3.06***	**3.41***	**3.39***
Playful	5.27	**4.15***	**4.03***	4.04*
Pleading	2.90	2.36	2.45	2.51
Skeptical	4.49	**2.90***	**3.36***	**3.04***
Sorry	1.85	**1.49***	**1.65**	**1.58***
Surprised	5.66	**4.03***	**4.20***	**4.08***
Thoughtful	2.70	2.53	2.64	2.54
Narrative	4.94	**3.89***	**3.91***	**3.89***

6.4.2.3 Phrasing Error

Phrasing is decisive in perceptual judgements, as we emphasized in Sections 3.2.3 and 5.4.2. Notably, varying frequency of silences when modifying the speaking rate is a key feature of natural voice that synthetic models generally struggle to achieve. The pause embedding bias control developed in Section 5.4.1 was not used in the current experiment. However, note that the LST module enables local embedding biases to be added specifically at word boundaries. This bias addition would be able to simulate the pause embedding bias if any improvements of the training losses were found through the encoding of pauses in one or more local tokens. More detailed analysis of the local tokens used at word boundaries is needed to assess this behavior.

Table 6.5 shows mean silence proportion per style for each model and ***GT***. Significant differences between ***GT*** and synthetic models for "Pleading", "Skeptical", "Sorry", and "Narrative" demonstrate the difficulties of the TTS model to replicate natural balance between speech and silence for these styles. The LST module does not provide much improvement in that regard. Conversely, ***LST$_P$*** produces more pauses for styles with high silence ratio like "Angry" and "Playful", whose natural behaviors are hardly replicated by utterance-wise style bias in ***GST*** (this improvement was significant for "Angry").

Duration modulation was also evaluated as an indicator of local prosodic patterns. We hypothesize that polysyllabic words should be more impacted by local modulations, as they are mostly content words. At least some of the studied styles should emphasize local key points in the utterances that are embodied by content words. Word duration modulation is evaluated

Table 6.5: Mean proportion of silences in synthetic vs. ***GT*** utterances (in %). ** indicates that distributions statistically differ from the ***GT*** with p < 0.01. Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the ***GST*** baseline.

Style	*GT*	*GST*	LST_W	LST_P
Angry	2.6	**2.0****	**1.8****	**3.0**
Comforting	2.5	2.3	2.2	1.9*
Committed	4.8	3.3	3.4	3.3
Enthusiastic	1.6	1.7	1.6	1.2
Obvious	0.8	1.1	0.8	1.0
Playful	6.6	4.7	4.8	5.2
Pleading	1.3	**0.6****	**0.6****	**0.7****
Skeptical	2.4	**1.8****	**2.0****	**1.5****
Sorry	2.2	**1.4****	**3.2****	**1.9****
Surprised	2.0	1.9	1.4	1.0
Thoughtful	1.8	2.4	1.4	1.6
Narrative	3.8	**2.5****	**2.6****	**2.6****

as the ratio between the duration of the last vowel and the mean duration of other vowels of the same word. This measure indicates the lengthening of last syllable of polysyllabic words, as approximation of content words. Table 6.6 summarizes the evaluated duration modulation per style. Lengthening of the last syllable of polysyllabic words is very common in ***GT***, as shown by mean word duration modulations above 1.25 for every style. "Obvious", "Pleading", "Sorry" and "Thoughtful" show the higher degree of modulation. This modulation is closely replicated by all models, with slight variations between them. Interestingly, ***GST*** tends to elongate durations excessively, in particular on "Enthusiastic", "Playful" and "Thoughtful", while the LST modules help producing more natural duration modulations.

6.4.3 Perceptual Evaluation

In order to evaluate perceptual differences between the proposed model and the baseline, 60 participants took part in an online MUSHRA-like experiment [International Telecommunications Union, 2003], run with the framework webMUSHRA [Schoeffler et al., 2018]. Given the text uttered and the target style written for each utterance, participants were asked to evaluate on a scale from 0 (very bad) to 100 (excellent) if the style was correctly rendered. For this listening test, we selected 10 utterances per style that maximize spectral distances between systems (120 in total). 5 groups of 12 participants each evaluated 24 utterances (2 per style), with 5 systems per utterance: the ***GST***-enhanced FastSpeech2 baseline, the two proposed models LST_W and LST_P, the vocoded ***GT*** (hidden reference), and a FastSpeech2 trained without GST on non-expressive data (low anchor) referred to as ***LA***. Because the Ground Truth is not the only way to convey the given style, it was not given as an explicit

Table 6.6: End syllable duration modulation evaluated on polysyllabic words. * indicates that the distribution statistically differs from the ***GT*** ($p < 0.05$). Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the ***GST*** baseline.

Style	***GT***	***GST***	***LST$_W$***	***LST$_P$***
Angry	1.34	1.34	1.28	1.34
Comforting	1.33	1.35	1.39	1.36
Committed	1.34	1.34	1.39	1.34
Enthusiastic	1.36	1.47	1.44	1.38
Obvious	1.41	1.43	1.41	1.40
Playful	1.32	1.54	1.44	1.51
Pleading	1.37	1.30	1.35	1.30
Skeptical	1.24	1.28	1.31	1.22
Sorry	1.43	1.44	1.52	1.46
Surprised	1.25	1.23	1.27	1.21
Thoughtful	1.88	1.95	1.91	1.95
Narrative	1.33	**1.36***	**1.36***	**1.37***

Table 6.7: MUSHRA-like score per style. Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the ***GST*** baseline. * and ** indicates that this difference with ***GST*** is statistically significant with $p < 0.05$ and $p < 0.01$, respectively. **LA** = Low Anchor, ***GT*** = Ground-Truth.

Style	**LA**	***GST***	***LST$_W$***	***LST$_P$***	***GT***
Angry	17.3	63.0	64.3	**68.3****	75.6
Comforting	15.4	66.2	63.5	61.5	80.5
Committed	24.9	65.1	**70.9****	68	76.4
Enthusiastic	11.6	66.2	70.0	**74.0***	86.4
Obvious	40.2	65.7	61.4	65.3	84.7
Playful	16.4	63.3	66.3	67.4	86.5
Pleading	12.3	71.3	70.1	71.2	77.9
Skeptical	36.3	47.3	50.6	46.6	63.3
Sorry	15.4	63.2	**71.1****	68.0	68.7
Surprised	14.3	78.5	75.6	73.7	85.3
Thoughtful	24.3	46.9	47.5	52.7	62.7
Narrative	64.6	63.1	**67.4***	**67.5***	69.5
Total	24.2	63.0	64.7	65.0	76.1

reference to the participants during the listening test. Participants who misunderstood the evaluation task were excluded: it includes ranking the non-expressive model higher than the other models, as well as participants with significantly low standard deviation of grades across all systems. Examples rated by participants can be found at the following link[5].

Results of this perceptual experiment are given in Table 6.7. ***LA*** was ranked significantly lower than all other models, except for "Narrative" which is also modeled by the non-expressive ***LA***. Participants tend to favor $\boldsymbol{LST_W}$ and $\boldsymbol{LST_P}$ over ***GST***. Most noticeable improvements are found for "Angry", "Committed", "Enthusiastic", "Sorry" and "Narrative". Objective evaluations have shown that the LST module helps producing local behaviors that are closer to the ***GT***. Reproducing pitch variations and phrasing is critical for these styles to be perceived as natural. Note that ***GT*** exhibited relatively poor results on "Skeptical" and "Thoughtful". These styles may have been too caricatured by the speaker, which participants judged as unnatural. These performances may also be explained by the mislabeling of these styles, either because of the misinterpretation of the target styles by the speaker during recording sessions, or the diversity of meanings of these labels which ultimately impacted participants' judgments.

6.5 More Specific Annotation of the Expressive Recordings

As discussed in Section 1.2, expressive labeling is hard due to the variability of produced and perceived expressive style [Bachorowski, 1999]. In this work, expressive labels are taken as the target given to the speaker during recording sessions. We are fully aware that these labels may be inaccurate and the expressive style produced by the speaker may not be widely recognized by naive listeners. The speaker may have played her own understanding of the style requested. The production may also not be consistent throughout an entire session, either due to loss of focus or fatigue.

To evaluate the produced expressive style more precisely, a web interface was developed to annotate the recorded corpus through crowdsourcing. This project is called Emotags [Bailly et al., 2024][6]. This interface is illustrated in Fig. 6.4. Participants are given three ways to annotate the video clips:

1. A limited list of 6 to 12 tags determined incrementally through navigation in a large set of 132 expressive adjectives.
2. Free selection in the large set via an autocomplete text input.
3. A free text input if the label does not already exist in the large set.

The annotation process is ongoing at the time of writing of this manuscript[7]. 438 participants have already annotated 8547 utterances, with at least 2 expressive styles by utterance.

[5] https://www.gipsa-lab.grenoble-inp.fr/~martin.lenglet/listening_page_LST/index.html

[6] I have contributed to the experiment design and to the analysis of the collected annotations. The implementation of the website was done by Romain Legrand and Gérard Bailly.

[7] Link to the experiment: https://expe.univ-grenoble-alpes.fr/emotags/?PROLIFIC_PID=theradia&STUDY_ID=gipsa&SESSION_ID=1

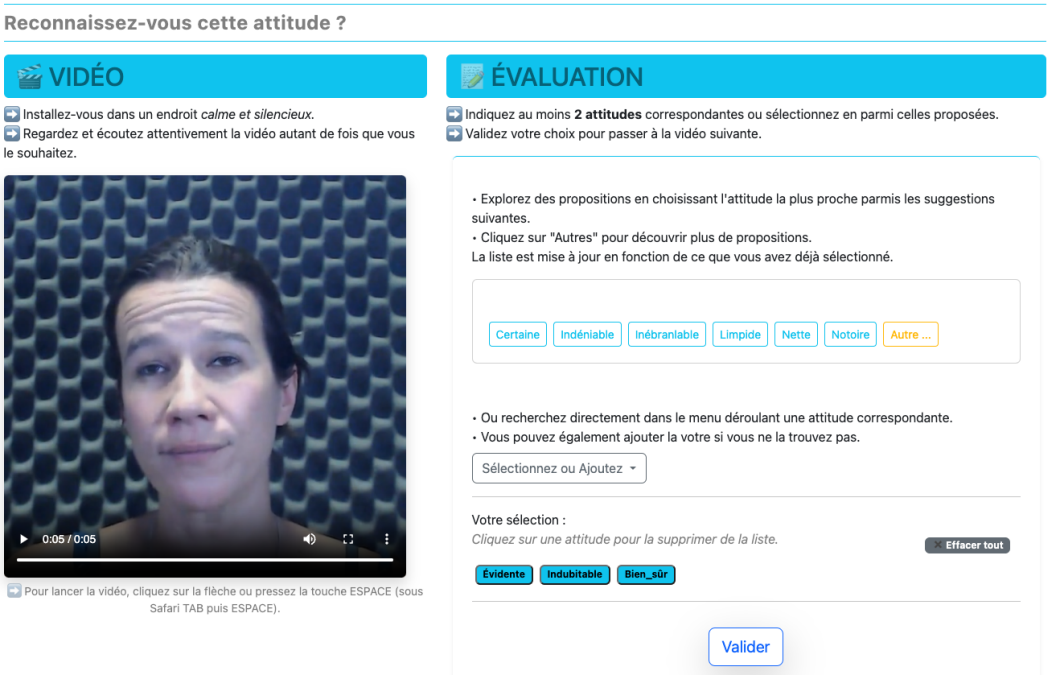


Figure 6.4: Screenshot of the web interface designed to annotate the recorded expressive corpus.

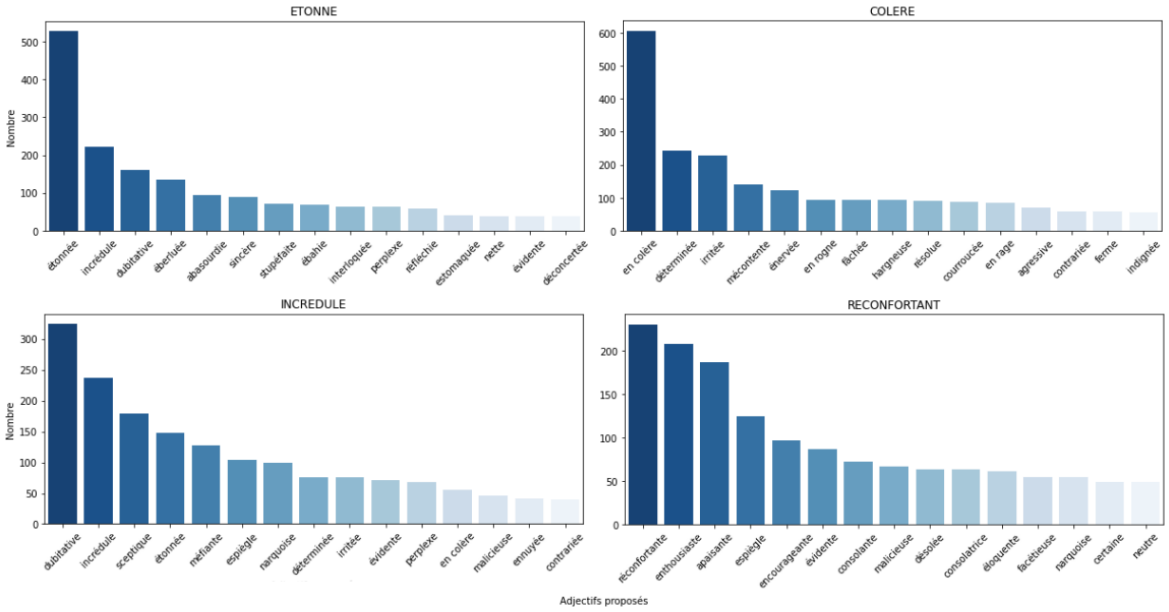


Figure 6.5: Annotations collected through the Emotags Interface for attitudes "Surprised" (ETONNE), "Angry" (COLERE), "Skeptical" (INCREDULE) and "Comforting" (RECONFORTANT).

More than 1000 participants are expected to annotate the whole expressive corpus. 65 534 have already been collected through this large-scale crowdsourcing. Preliminary results of this experiment are given in Fig. 6.5. The Ground Truth label is the most given label for all attitudes except "Skeptical" ("Doubtful" is the most given attribute in this case). This lack of recognition may have impacted the scores reported in the MUSHRA (see Table 6.7). For other

attitudes, this experiment mostly confirms that our speaker has successfully transmitted the intented attitude, but nuances may vary within the utterances from the same attitude.

We hope that such massive gathering of expressive annotations will enable us to 1) refine the training of an expressive TTS model on more accurate style labels as well as 2) train an expressive TTS model on a wider variety of expressive control, using pre-trained representations from Large-Language-Models such as FlauBERT [Le et al., 2020] to introduce a free text-control following the framework established by M. Kim et al. [2021] and Shin et al. [2022]. The training of such a model adapted for French TTS is left for future work.

6.6 Discussion

In this chapter, we proposed the LST module for expressive TTS which helps with modeling fine-grained prosodic patterns. This auxiliary module acts as a residual layer which complements the global style control achieved by the GST layer with contextual syntactic and semantic information. No additional losses are required to train this LST module. This module was evaluated on 12 common attitudes, for French synthesis. The most promising improvements over the GST baseline were shown for "Angry", "Committed", "Enthusiastic" and "Sorry", for which more subtle prosodic variations are needed to achieve a natural behavior.

The LST module was implemented at word and phone scales in this work. Both scales showed improvements in the perceptual test (see Table 6.7), but on different styles. This may indicate that various levels of prosodic contribution are needed to model different styles. "Committed" and "Sorry" may rely more on emphasis produced on the right words or groups of words given by the structure of the text input, whereas the natural rendering of "Angry" and "Enthusiastic" needs more intra-word modulations. The structure of the proposed LST module allows for several LST modules to be cascaded to encode increasingly finer representations such as phrases, words, syllables, phones, etc. We will consider the stacking of various scales of local embeddings in future work.

6.6.1 Evaluation of Local Contributions

This study reinforces the need for more elaborated evaluation paradigms for expressive speech. While the style "Sorry" showed the highest amount of objective errors compared to the Ground Truth, it was still perceived as well-rendered during listening tests. Prosodic patterns followed by the Ground Truth are not exclusive, and evaluation has to be adapted to match perceptual judgments. We tried to adapt our objective evaluation in that regard, by comparing the distribution of synthetic acoustic features with the Ground Truth measurements instead of one-to-one phone error computation. However, state-of-the-art expressive modules for TTS like GST already achieve such high performance that most predicted features are already very similar to the Ground Truth. That is why evaluation setups should now increasingly consider outliers and extrapolations out of their typical operating range, on which neural models still have room for improvements [Perrotin et al., 2021].

The MUSHRA-like setup we used in this subjective experiment did not provide enough insights into the perceptual cues upon which participants based their judgments. As illustrated in Fig. 6.2, the LST module contributes to subtle short range variations of prosodic features. These variations seem to be perceived by listeners as showed by the difference in scores between GST and LST modules, but utterance-scale objective measurements fail to capture these events. We will consider using the Rapid Prosody Transcription (RPT) evaluation framework for future works in order to isolate the shorter acoustic windows which are decisive in participant judgments.

6.6.2 Acoustic Probing in Local Tokens

The number of tokens and training process of the LST module deserves more attention. The best results were found for styles that make use of multiple local tokens (Table 6.2 and Fig. 6.3). This result was expected, since adding the same local token all along the utterance should not provide different results from an utterance-wise style bias. Constraining the LST module to maximize token usage should help the model showing more robust results. Additionally, the number of local tokens should be adapted to the scale of representations, e.g. allowing more various contributions for finer-grained prosodic patterns.

Understanding the features encoded in local tokens would also help in choosing the appropriate number of tokens. The acoustic probing methods proposed in Chapter 4 could be applied to the local prosodic representations computed by the LST module. This analysis is expected to show how the usage of individual local tokens correlates with the acoustic bias applied to the local speech units. This analysis would also highlight acoustic and prosodic similarities and divergences between styles, in a new attempt to take advantage of statistical learning to contribute to the better understanding of speech mechanisms.

6.6.3 Integration of Large Language Model Representations

The LST is proposed as an auxiliary module to learn the entanglement between the linguistic structure of the utterance and the global propositional attitude to convey. This assumption relies on the ability of the text encoder of neural TTS to model syntactic and semantic information. This information is partially encoded into latent embeddings, as shown by the partial disambiguation of homographs evaluated in Section 2.4.3. However, the primary focus of neural TTS is the prediction of spectral features, which do not require a deeper understanding of the textual content in most cases. On the other hand, Large Language Models (LLMs) like BERT [Devlin et al., 2018] are trained on language prediction tasks on larger corpora. This setup has been shown to favor the encoding of contextualized semantic information [Wiedemann et al., 2019]. We believe that this additional contextual information would benefit the LST local prosodic prediction. Future work should explore the combination of such pre-trained contextualized representations with the LST mechanism.

Chapter 7

Application to Audio-Visual Synthesis

Contents

Chapter Highlights

This chapter presents our **Audio-Visual Text-to-Speech (AV-TTS)** model based on the FastSpeech2 architecture. Our model generates the speech and co-verbal gesture of the **Embodied Conversational Agent (ECA)** of the Theradia application. The proposed AV-TTS combines an audio-visual Transformer-based encoder with two distinct audio and visual neural decoders that generate expressive speech from partially-annotated data. We show that this model is able to produce **recognizable expressive behaviors** for the ECA.

As part of the Theradia project [Tarpin-Bernard et al., 2021], our contributions proposed in the previous chapters were integrated into an audio-visual generative model which predicts both speech and co-verbal facial gestures from text. This joint prediction is transformed into the animation of an embodied conversational agent (ECA), which aims to accompany patients who are undergoing online digital therapies.

Theradia's desire to embody the voice of a virtual caregiver in an ECA aligns with the broader goal of providing a more interactive experience to its patients. Co-verbal gesture helps to create more engaging interactions [Nyatsanga et al., 2023], provided that these gestures are consistent with its verbal behavior [McNeill, 2019] and with the active listening capabilities of the avatar [Potdevin, 2020]. Therefore, the Audio-Visual generation from text (called **AV-TTS**) may be considered as an extension of the expressive TTS framework, as it undergoes the same problematic of linguistic and paralinguistic content modeling, but includes the visual modality as output.

Several state-of-the-art reviews of AV-TTS systems and architectures have been published for concatenative and statistical systems (see [Bailly et al., 2003; Mattheyses & Verhelst, 2015; Theobald, 2007]). To the best of our knowledge, no such review is available for neural architectures.

The latest propositions of unifying speech and gesture generative models [S. Wang et al., 2021] into a single neural AV-TTS have allowed the emergence of jointly generative models with remarkable performance [Hussen Abdelaziz et al., 2021; Mehta et al., 2023; Yu et al., 2020]. Notably, DurIAN [Yu et al., 2020] and AVTacotron2 [Hussen Abdelaziz et al., 2021] have extended the TTS-architecture from Tacotron and Tacotron2, respectively, in order to generate visual features directly from audio-visual representations computed from text. In both models, visual features are predicted from the very end of the autoregressive decoder. This setup ensures the computation of audio-visual representations in the whole model. Although we also believe that the computation of audio-visual representations should benefit the consistency of generated synchronous features, the visual modality may exhibit some asynchronous behavior such as anticipatory voice actuator activation [Maier et al., 2011]. Therefore, we consider that an earlier distinction between the audio and visual decoders should benefit both modalities.

Dahmani et al. [2019, 2021] proposed to pre-train separate audio vs. visual encoder-decoder VAE models biased by phone labels and durations so that the audio and visual decoders can be further used by a text encoder. But to the best of our knowledge, we propose in this chapter the first end-to-end expressive AV-TTS based on the FastSpeech2 architecture [Ren et al., 2021]. We combine an audio-visual Transformer-based encoder with two distinct decoders. Following the FastSpeech2 framework, we integrate a visual variance adaptor to the visual decoder, in order to better model lip movement.

Section 7.1 presents the multi-modal and multi-speaker corpus used to train our AV-TTS. The AV-TTS architecture is further detailed in Section 7.2. We evaluate the expressive capabilities of our model in Section 7.3.

7.1 Audio-Visual Multi-Speaker Corpus

The model presented in this chapter has been trained with the goal of being integrated into the application Theradia [Tarpin-Bernard et al., 2021]. Thus, the entire corpus described in Appendix A is used for the training, in order to maximize the expressive capabilities of the model. This corpus includes voices from five speakers, for a total duration of 51:28:39. Only two speakers have been recorded with visual settings (AD and IZ), and only one with expressive labels (AD). The recording of the expressive dataset, as well as the visual features extraction from the videos and their use to animate the virtual agent are detailed in Appendix A. In this section, we use the reduced set of 37 **Action Units (AU)** as target for the visual decoder. The data sparsity of the corpus is handled during the training, as described in Section 7.2.2. Although transfer learning is expected to enable the use of expressive labels for other speakers than AD and the prediction of AU for speakers who have not been recorded with visual settings, the 3D model of the avatar has been fine-tuned on AU from AD. Thus only AD is included in the test set of the presented experiment. This test set is the same as that presented in Table 6.1.

113 164 <orthographic|phonetic> pairs (without audio transcripts) complement this corpus. These transcriptions are used to train the text encoder and the phonetic predictor on a wider variety of linguistic contexts, as described in Section 2.3.2. Both orthographic and phonetic inputs are used during training.

7.2 Audio-Visual Generation from Text

The proposed AV-TTS is illustrated in Fig. 7.1. This model is based on the LST-enhanced FastSpeech2 model presented in Chapter 6. The constrained-GST encoder (Fig. 7.1, label 1) computes an utterance-wise style embedding from an audio reference which is added to the output of the text encoder. Trainable speaker embeddings (Fig. 7.1, label 2) are also added at the output of the encoder. The LST module (Fig. 7.1, label 3) proposed in Section 6.3 further modulates the biased sequence of text-embeddings at the word-level. We favor the word-level for its expected ability to model semantic information, in comparison with phone-level. The phonetic predictor (Fig. 7.1, label 4) proposed in Section 2.3.2 is implemented at the output of the LST module. The biased sequence of character/phone embeddings is cloned, and transmitted to the Audio Decoder and the Visual Decoder (Fig. 7.1, label 5). This cloning does not stop the gradient propagation, therefore the text encoder is constrained to produce audio-visual embeddings, as advocated by S. Wang et al. [2021].

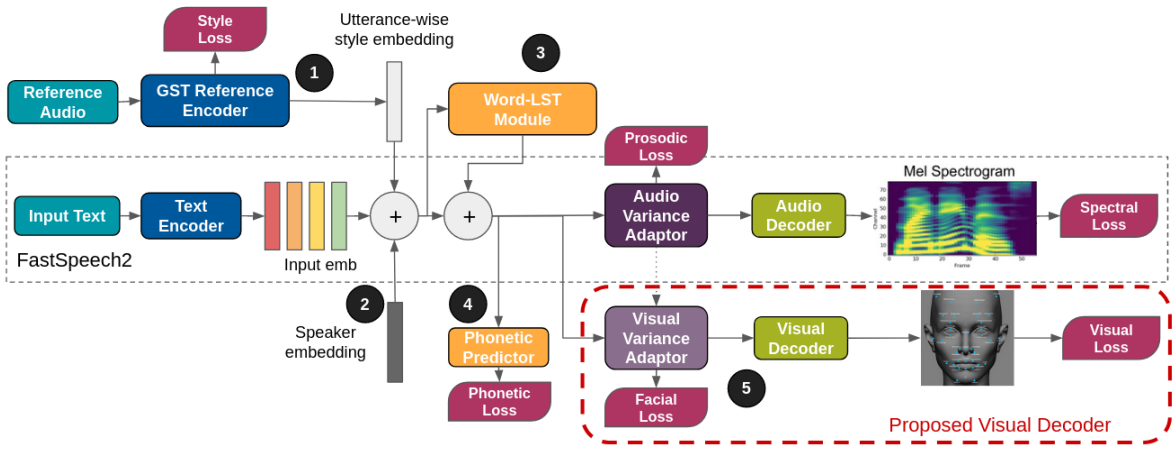
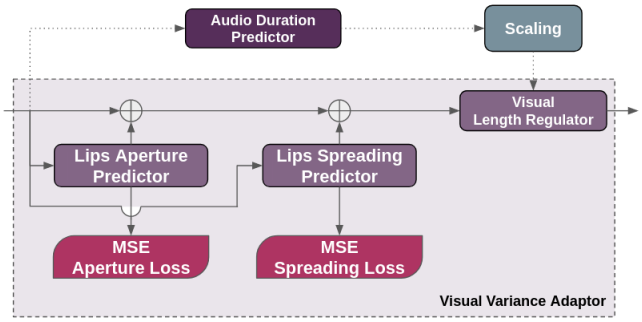


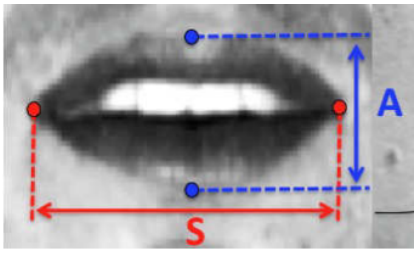
Figure 7.1: Proposed Audio-Visual FastSpeech2 model. The visual variance adaptor is illustrated in Fig. 7.2a.

7.2.1 Visual Decoder

The visual decoder follows the same architecture as the audio decoder: a variance adaptor followed by four stacks of FFT blocks (see Fig. 1.5a) and a fully connected layer to project the output of the last FFT block onto the 37 AU. The visual variance adaptor is illustrated in Fig. 7.2a. Similar to the audio variance adaptor from FastSpeech2 [Ren et al., 2021], the visual



(a) Visual Variance Adaptor: two predictors are implemented for lips aperture and width respectively. Note that the visual variance adaptor does not implement a duration predictor, but instead uses the duration predicted by the audio duration predictor.



(b) Lip Aperture (A) and lips spreading (S). Source: [Garnier et al., 2012]

Figure 7.2: Description of the Visual Variance Adaptor.

variance adaptor implements the explicit prediction of two visual features: the lip aperture (A) and spreading (S) as illustrated in Fig. 7.2b. The lips stand as a primary position in the face-to-face interaction: poor lip sync has been shown to impact intelligibility [McGurk & MacDonald, 1976]. The modeling of lip aperture and spreading is therefore an additional constraint to avoid conflicts between the visual and the audio modalities. The implementation of these lip predictors at the output of the audio-visual encoder takes advantage of the synchronicity between the linguistic content and the lips modeling.

Both predictors are trained with a MSE loss, which ensures the encoding of these two features in the audio-visual embeddings computed by the encoder. Each predictor computes a scalar value for A and S respectively, which is converted into an aperture embedding and a spreading embedding added to the corresponding character/phone embedding. The total loss of our proposed AV-TTS is given by formula 7.1. Note that Ground Truth values of lip aperture and spreading are computed as a linear combination of the 10 AU dedicated to the animation of the lips, since these features are not explicitly encoded by the AU.

$$L_{FS} = L_S + L_{PS} + L_{dur} + L_p + L_e + L_{phon} + L_V + L_{PV} + L_{LA} + L_{LS} \tag{7.1}$$

with L_{FS} the total loss of FastSpeech2, L_S the MAE spectral loss, L_{PS} the MAE spectral loss after the postnet, L_{dur} the MSE duration loss, L_p the MSE pitch loss, L_e the MSE energy loss, L_{phon} the cross-entropy phonetic loss, L_V the MAE visual loss on AU, L_{PV} the MAE visual loss after the postnet, and L_{LA} and L_{LS} the MSE lip aperture and spreading losses respectively.

Note that the aperture and spreading embeddings (resp. pitch and energy embeddings) are only added to the embedding sequence of the visual decoder (resp. audio decoder). This avoids the overwriting of encoded features due to the summing of variance embeddings as observed after the summing of pitch and energy embeddings in Fig. 4.2a for $\boldsymbol{FS}$ and $\boldsymbol{FS}_{\backslash \boldsymbol{phon}}$.

We did not implement a duration predictor in the visual variance adaptor. Training two duration predictors may cause asynchronicity at inference between the audio and visual modalities. Instead, only the audio duration predictor is trained. The audio duration predictor is trained to predict the number of mel-spectrogram frames to produce from each element of the input sequence. On the one hand, with the mel-spectrogram configuration given in Table D.1, the mel-spectrogram is predicted with a temporal sampling-frequency of $\sim$86 Hz. On the other hand, AU are predicted with a sampling-frequency of 60 Hz. Thus, the number of frames predicted by the audio duration predictor is down-scaled by a factor $60 \div 86 \approx 0.7$.

7.2.2 Handling Training Data Sparsity

This model is trained to predict three different outputs (the mel-spectrogram, the visual features and the one-to-one phonetic mapping) from three types of inputs (the input text, the audio reference and the style label). In practice, any of these parameters, text input excluded, may be missing in the training corpus: only two speakers have been recorded with audio-visual settings, dictionary input transcriptions are missing the output spectrogram, as well as the reference audio, and expressive labels are only available for one speaker.

However, the model's architecture enables the training on any minimal <input|output> pair[1]. In case of missing audio recording, the GST-Reference Encoder is ignored and the Style Loss is not computed on this sequence. As a result, the style embedding vector is a zero-vector, and does not contributes the prediction. Only the phonetic output is predicted, as the audio and visual decoders are by-passed in absence of target. Thus, only the text encoder and the phonetic predictor are trained in this case. Similarly, in absence of visual features (resp. expressive labels), the visual decoder (resp. the style loss) is by-passed.

Training on sparse datasets necessitates special attention when it comes to data mixing in batches during the training. Since each type of minimal <input|output> pair only trains a portion of the model, unbalanced types of data in the training dataset may favor the minimization of one loss compared to the other. That is why we advocate for the training procedure described in Appendix C: we empirically found that a ratio of 2/3 of audio inputs and 1/3 of non-audio inputs in each batch provides the best balance between phonetic prediction accuracy and spectral loss minimization. Because of the small portion of visual data in our corpus (1/5 of the dataset), we cannot afford this ratio between audio only and audio-visual data without overfitting the model on AD, but similar caution should be considered with wider corpora.

7.3 Evaluation

7.3.1 Experimental Setup

The multi-speaker training of the proposed AV-TTS follows the procedure described in Appendix C. We use HiFi-GAN as a vocoder, with the configuration described in Appendix D.3.

[1] At least one input and one output.

Due to time limitations, we did not train any baseline AV-TTS. With additional time, we would consider comparing the performance of our proposed model to the AV-Tacotron2 model proposed by Hussen Abdelaziz et al. [2021] which follows the same global architecture.

We provide some examples of the facial expressions generated by our model in Fig. 7.3. Note that the control provided by the ECA enables the model to exhibit basic facial expressions and head postures, but does not integrate the control of arms. "Enthusiastic" shows a large smile (Fig. 7.3b). "Thoughtful" looks away as if the ECA was looking for its words (Fig. 7.3c). When "Surprised", the ECA's eyes widen (Fig. 7.3d). These behaviors replicate the acted behaviors of the speaker during recordings.

(a) "Narrative" (b) "Enthusiastic" (c) "Thoughtful" (d) "Surprised"

Figure 7.3: Examples of facial expression generated by the AV-TTS.

To assess our model expressive capabilities, we conducted two forced-choice recognition evaluations online: 1) using AU extracted from the Ground Truth to animate the avatar, and the original audio recordings, and 2) with audio-visual features predicted by the model. Each setup allows for the evaluation of one hypothesis:

H1: (Experiment 1) the accuracy of the AU *extracted from the video recordings*, and their use as control parameters for the 3D model of the ECA lead to a high recognition of our 12 attitudes.

H2: (Experiment 2) the accuracy of the AU *predicted by our model*, and their use as control parameters for the 3D model of the ECA lead to a high recognition of our 12 attitudes.

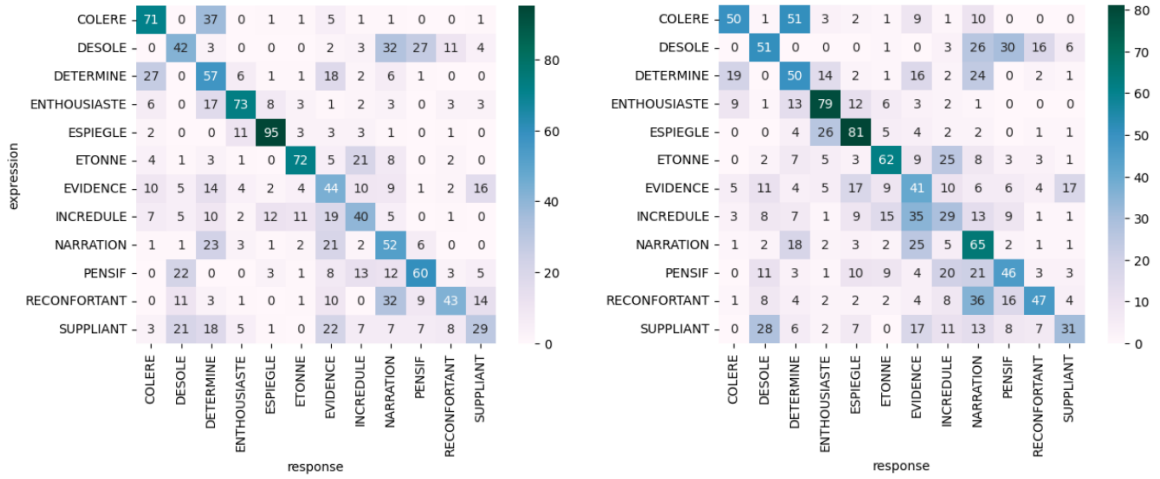
In order to focus the evaluation on distinctive examples, we selected the test set as the most varied synthetic utterances between attitudes: we computed the pair-wise MSE between predicted spectra by attitude ($12 \times 11 \div 2 = 66$ pairs by utterance). Predicted spectra were previously aligned with DTW. These pair-wise errors are averaged by utterance, before selecting the 20 most varied utterances. The syntheses of these 20 utterances with the 12 attitudes (240 stimuli) were used as test set for both experiments.

49 participants recruited on Prolific [Palan & Schitter, 2018] took part in the experiments. Each participant evaluated 60 stimuli, randomly mixed between experiments 1 and 2. Participants were presented with the video of the animation of the avatar, and were allowed to play this video as many times as necessary. Then they had to select one of the twelve labels before continuing to the next video.

7.3.2 Results

The result of the recognition experiments are given in Fig. 7.4a for Ground-Truth features and Fig. 7.4b for predicted features. The F1-score calculated from the confusion matrix is 0.47 for Fig.7.4a and 0.41 for Fig.7.4b. We computed the cosine similarity between the two matrices as an indicator of the correlation between the attitude recognition on the Ground Truth features and the predicted features. We obtain a correlation score of 0.92 between the two confusion matrices.

The most common confusions found in both matrices are summarized in Table 7.1. Most of these confusions may be explained by acoustic similarities between the two attitudes. The acoustic evaluation of the Ground-Truth recordings for the speaker AD in Chapter 6 indicates that "Angry", "Committed", "Obvious", "Skeptical" and "Surprised" produce higher F0 intra-utterance variations (see Fig. 6.4). Conversely, "Sorry" and "Comforting" show the least amount of F0 variations. "Pleading" and "Obvious" show relatively high end-syllable duration



(a) Ground-Truth Visual Features

(b) Predicted Visual Features

Figure 7.4: Confusion matrices of attitude recognition using visual features (AU) from the Ground Truth (left) and predicted visual features (right).

Table 7.1: Common confusions between attitudes. Note that these confusions are symmetrical.

Style 1		Style 2		Acoustic Similarity
En	**Fr**	**En**	**Fr**	
Sorry	Désolé	Thoughtful	Pensif	slow speaking rate
Sorry	Désolé	Comforting	Réconfortant	slow speaking rate, lower mean F0, less F0 variations
Committed	Déterminé	Angry	Colère	more frequent breaks (emphasis), high F0 variations
Committed	Déterminé	Obvious	Évidence	high F0 variations
Pleading	Suppliant	Obvious	Évidence	end-syllable lengthening
Skeptical	Incrédule	Surprised	Étonné	high F0 variations, semantic

lengthening, though not as high enough to be confused with "Thoughtful" (see Fig. 6.6). The pause frequency shown in Fig. 6.5 indicates that "Angry" and "Committed" lead to frequent pauses. These pauses are mostly used as markers of emphasis for the following words, which is a common speech behavior in both these attitudes. Although speaking rate was not explicitly measured in Chapter 6, the speaker slows her speaking rate when producing "Sorry", "Thoughtful" and "Comforting", which may have introduced further confusion between these attitudes. The confusion between "Skeptical" and "Surprised" is reinforced by the semantic proximity between these two adjectives: skepticism may be expressed through surprise, which may have perplexed participants.

These acoustic similarities should have been mitigated by the associated differences of facial expressions. For example, the speaker produces "Angry" utterances with pronounced furrowed brows, in opposition to a more friendly face for "Committed". These nuances of facial expressions are likely not transmitted to the ECA, even in the Ground-Truth condition (F1-score of 0.47 in Fig. 7.4a). This mostly invalidates **H1**, as the control of the ECA may be too restrictive to transmit the appropriate nuances of facial expressions. Nonetheless, the cosine similarity of 0.92 between the two confusion matrices of Fig. 7.4 indicates that AU are learned by the proposed AV-TTS in accordance with the Ground-Truth recordings, which confirms **H2**.

Note that the two confusion matrices of Fig. 7.4 also indicate asymmetrical recognition errors on "Narrative". Interestingly, "Narrative" was confused as "Obvious" and "Committed", while this confusion does not happen in the other direction. Conversely, "Sorry" and "Comforting" have been recognized as "Narrative". "Narrative" may have been a misleading label to evaluate isolated speech extracts, as "Narrative" more often implies longer forms of speech like story telling. "Neutral" may have introduced less confusion.

7.4 Conclusions and Discussion

In this chapter, we introduced our FastSpeech2-based expressive AV-TTS trained on partially-labeled expressive data. We showed that this model was able to produce expressive speech and co-verbal gesture for an ECA, used in the scope of the Theradia project [Tarpin-Bernard et al., 2021].

From the evaluation presented in Section 7.3, we showed that our proposed AV-TTS produced speech and visual features that are very similar to the features extracted from the Ground Truth recordings particularly in terms of how they are perceived by naive observers in recognizing attitudes. Therefore, the AV-TTS system for joint automatic speech and animation generation is capable of capturing the patterns specific to the chosen attitudes. However, the poor recognition of the Ground Truth (Fig. 7.4a) features may indicate that:

- The speaker AD has not produced consistent attitudes during recording sessions. This may be due to fatigue or a mismatch between listener expectations and speaker production. However, perceptual results of the Ground Truth samples presented in Table 6.7

indicate that (at least audio) recordings have been evaluated as good to excellent[2] renditions of the corresponding attitudes ("Skeptical", "Sorry" and "Thoughtful" being the least well rendered through the audio modality).

- The choice of attitude labels may have introduced some confusion. "Skeptical" and "Surprised" may be expressed in the same contexts, and are thus harder to distinguish on isolated utterances. The preliminary results from Emotags (see Section 6.5) suggest that "Skeptical" ("Incrédule" in French) could be replaced by "Doubtful" ("Dubitatif" in French). "Narrative" may also be hard to judge on isolated utterances.
- The visual feature tracking and control of the avatar need improvements. As illustrated in Fig. 7.3, the visual control of the ECA is limited to basic facial expressions and head tilt. The control of the body posture and arm movements may help to model some of the presented attitudes. However these features have not been recorded in the current setup and are not implemented in the ECA 3D model.

Due to the lack of time, we have not evaluated our proposed model in comparison with similar AV-TTS architectures. The comparison with AVTacotron2 [Hussen Abdelaziz et al., 2021] would provide interesting insights into the benefits of our proposed Transformer-based architecture compared to recurrent TTS.

Among the perspectives for improvements, we consider extending the visual variance adaptor to visual events, such as eye blinks and head nods. In the current version of our model, these events are not replicated, because of their infrequent occurrence in the training dataset. The variance adaptor could be adjusted to estimate the likelihood of initiating these events, which, in turn, would trigger pre-recorded sequences to replace the predicted eye or neck movements during inference. Another avenue for improvement is the integration of the visual modality as input of the GST reference encoder. The role of the GST encoder in our architecture is to model the paralinguistic content not yet explained by the text input. Gesture is part of the paralinguistic content, and thus should benefit the modeling of the utterance-wise attitude embedding.

[2]Labels from the corresponding MUSHRA scale: [60-80] is "Good", [80-100] is "Excellent".

Chapter 8

Conclusions and Perspectives

This chapter draws general conclusions about the various contributions made in this thesis. The main research outcomes are discussed in Section 8.1. Directions of future work are finally discussed in Section 8.2.

8.1 Main Contributions

In this thesis, we tackled the modeling of expressive speech through an innovative approach that dissects the workings of neural Text-to-Speech (TTS) models. This approach stands out from the current trend of proposing increasingly complex neural models that prioritize performance gains, often at the expense of the interpretability of the underlying learning processes. The opening of these neural black boxes according to the principles of Explainable Artificial Intelligence is not only presented here as a means to bridge the gap between language sciences and speech technologies but also a necessity for designing future models that leverage the knowledge accumulated in the literature.

The first step in that direction involved analyzing how the presentation of training data to the model influences the predicted synthetic speech. The two studies performed on Tacotron2 in Chapter 3 have confirmed the importance of modeling contextual information encoded in the linguistic content, both at the intra-utterance and inter-utterance levels. Notably, we highlighted that input segmentation into shorter utterances favored the accurate prediction of spectral features, at the expense of more natural phrasing when generating longer utterances. We attempted to compensate this flaw by the introduction of linking punctuation as explicit symbols to encode inter-utterance linguistic prosody, combined with the synthesis in smaller chunks. These results highlighted the limited capabilities of neural TTS to predict linguistic prosody solely from input text. Therefore, we emphasized the need to design control mechanisms that better respect the way these supra-segmental parameters are encoded by neural models.

In Chapter 4, we proposed our analytical method of linear probing of intermediate embeddings computed by neural TTS for this purpose. Our method reveals the underlying learning processes of neural TTS and enables the identification of the locations where acoustic and phonological features of interest are encoded. Notably, the identification of the features encoded by layer opens the route toward better informed design choices for TTS, with respect to the unsupervised learning processes of neural models. Our proposed method serves as a complementary approach to adversarial training, providing a means to validate the features encoded by neural embeddings. We designed this tracking method so that it can be adapted to any type of TTS architecture. We illustrated the potential of this approach on two of the main neural TTS frameworks at the time of writing of this thesis: Tacotron2 and FastSpeech2.

The linear probing by layer proposed in Chapter 4 was turned into a linguistic prosody control mechanism in Chapter 5. We demonstrated that the linear directions encoding acoustic and phonological features within latent embeddings could serve as explicit biases to control any of the identified features. Thus, we proposed a post-hoc control mechanism for continuous and categorical features, which does not require any additional data nor training of explicit predictors. This control has been verified on two architectures, but further testing is needed to assert its universality.

The better understanding of TTS embeddings enabled us to propose two neural modules:

1. First, the Local Style Token (LST) module proposed in Chapter 6 enabled us to modulate utterance-wise paralinguistic biases in order to produce more natural renditions of expressive attitudes. The LST serves as a bridge between linguistic and paralinguistic content, confirming that the production of propositional attitudes is anchored in the syntactic and semantic structure of the text.
2. Second, the FastSpeech2-based AV-TTS proposed in Chapter 7 extended the TTS framework to the generation of visual features to animate an Embodied Conversational Agent (ECA). To the best of our knowledge, this is the first attempt to jointly predict expressive speech and facial expressions using a FastSpeech2-based architecture.

8.2 Directions for future work

If space "disentanglement" of features is often sought for in controllable speech synthesis, each Chapter of this thesis has conversely shed light on the modeling of strong interplay between the numerous factors involved in speech production (linguistic and acoustic, linguistic and style, co-variation between acoustic features, local vs. global scales, etc.), and demonstrated that their exploitation can be far more beneficial for high quality synthesis than their removal. In this direction, several avenues for improvement have already been presented throughout this manuscript. Here, we revisit the main directions that emerge from this work.

8.2.1 Extension and Accessibility of the proposed Analytical Toolbox

The method presented, involving linear probing of features within intermediate embeddings, revealed intriguing insights into how information is encoded at the character- or phone-level of two of the main TTS architectures. We have emphasized that the proposed method is not model-specific, and thus could also be applied to other architectures. Notably, diffusion-based models are becoming increasingly important in the field of speech synthesis [Huang et al., 2022; Mehta et al., 2023; Popov et al., 2021]. Despite promising performance, the functioning of these models is not yet completely understood, which results in caution when considering wider applications. We believe that the proposed analytical methods may alleviate this issue. In this regard, future work should aim to simplify the analysis of models using our method, which currently involves a relatively complex process that requires the use of multiple languages and softwares (Python/MATLAB/Praat).

Additionally, I would like to extend the current analysis to other types of embeddings computed by neural TTS. A better understanding of speaker and style utterance-wise embeddings is expected to provide additional control mechanisms to extend model capabilities outside of their training corpus. I also aim to provide an acoustic analysis of the local tokens proposed in Chapter 6. This analysis is essential for interpreting the dynamics of local token usage in the context of understanding propositional attitudes as a sequence of local prosodic patterns grounded on the linguistic content.

8.2.2 Expressive Control with Free Style Tag

With regard to explicit expressive control, we believe that training on a limited set of expressive labels may not adequately capture the wide variability in speech production and perception. In contrast, the Emotags project [Bailly et al., 2024] (see Section 6.5) embraces this variability to provide hundreds of more nuanced labels. We aim to explore expressive control of neural TTS using free text labels, leveraging this collection of crowdsourced annotations as proposed by M. Kim et al. [2021] and Shin et al. [2022]. Our proposed analysis method would additionally provide a mixed interpretation of such style tag embeddings, both in terms of linguistic and acoustic features.

8.2.3 Better Evaluation Frameworks

The presented studies have all advocated for an evolution of the TTS evaluation framework. Overly-simplified evaluations of the naturalness of synthetic voices through MOS tests do not provide the insights needed to interpret the specific aspects that make new proposed models preferred or not. Perceptual tests need to be given particular care, with more explicit dimensions made comprehensible to naive participants.

Objective speech evaluations could also be improved. Future work should incorporate more specific objective analyses of portions of speech that are decisive in listeners' judgments. Methods like Rapid Prosodic Transcription (RPT) [Cole & Shattuck-Hufnagel, 2016] might be able to identify these salient segments for more focused acoustic analysis.

8.2.4 Deployed Application Monitoring

The AV-TTS presented in Chapter 7 has been implemented in the clinical version of the Theradia application. Cognitive remediation exercises with actual patients are thus accompanied by the generated ECA. The integration of the proposed model into a deployed application is the new challenge. Ongoing feedback from patients will guide the model's implementation choices to create an increasingly interactive and enjoyable experience.

Bibliography

First Author Publications (Attached to this document)

Lenglet, M., Perrotin, O., & Bailly, G. (2021). Impact of Segmentation and Annotation in French end-to-end Synthesis. *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 13–18 (cit. on pp. 61, 64, 65, 68).

Lenglet, M., Perrotin, O., & Bailly, G. (2022a). Modélisation de la Parole avec Tacotron2 : Analyse acoustique et phonétique des plongements de caractère. *Proc. XXXIVe Journées d'Études sur la Parole (JEP)*, 788–796 (cit. on pp. 39, 48).

Lenglet, M., Perrotin, O., & Bailly, G. (2022b). Speaking Rate Control of end-to-end TTS Models by Direct Manipulation of the Encoder's Output Embeddings. *Proc. Interspeech*, 11–15 (cit. on pp. 83, 99, 107, 111, 113, 118, 121, 122).

Lenglet, M., Perrotin, O., & Bailly, G. (2023a). A Closer Look at Internal Representations of End-To-End Text-To-Speech Models: How is Phonetic and Acoustic Information Encoded? *In preparation for a submission to Speech Communication*, 13–18 (cit. on pp. 83, 99).

Lenglet, M., Perrotin, O., & Bailly, G. (2023b). Local Style Tokens: Fine-Grained Prosodic Representations For TTS Expressive Control. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 120–126 (cit. on pp. xvii, 123, 124).

Lenglet, M., Perrotin, O., & Bailly, G. (2023c). The GIPSA-Lab Text-To-Speech System for the Blizzard Challenge 2023. *Proc. 18th Blizzard Challenge Workshop*, 34–39 (cit. on pp. 39, 53).

Collaborations

Bailly, G., Legrand, R., Lenglet, M., Elisei, F., Garnier, M., & Perrotin, O. (2024). Emotags: Computer-Assisted Verbal Labelling of Expressive Audiovisual Utterances for Expressive Multimodal TTS. *Submitted to LREC-Cooling* (cit. on pp. 123, 138, 155).

Bailly, G., Lenglet, M., Perrotin, O., & Klabbers, E. (2023). Advocating for text input in multi-speaker text-to-speech systems. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 1–7 (cit. on pp. 39, 52, 54, 56, 58).

Hajj, M.-L., Lenglet, M., Perrotin, O., & Bailly, G. (2022). Comparing NLP solutions for the disambiguation of French heterophonic homographs for end-to-end TTS systems. *SPECOM*, 265–278 (cit. on pp. 39, 54, 59, 173, 174).

Tarpin-Bernard, F., Fruitet, J., Vigne, J.-P., Constant, P., Chainay, H., Koenig, O., Ringeval, F., Bouchot, B., Bailly, G., Portet, F., Alisamir, S., Zhou, Y., Serre, J., Delerue, V., Fournier, H., Berenger, K., Zsoldos, I., Perrotin, O., Elisei, F., ... Ghenassia, D. (2021). THERADIA: Digital Therapies Augmented by Artificial Intelligence. *International Conference on Applied Human Factors and Ergonomics*, 478–485 (cit. on pp. 2, 143, 144, 150, 173–175, 187).

Other Citations

Adda-Decker, M., Boula de Mareüil, P., & Lamel, L. (1999). Pronunciation variants in French: schwa & liaison. *Proc. XIVth International Congress of Phonetic Sciences*, 2239–2242 (cit. on p. 7).

Adigwe, A. O., & Klabbers, E. (2022). Strategies for developing a Conversational Speech Dataset for Text-To-Speech Synthesis. *Proc. Interspeech*, 2318–2322 (cit. on p. 9).

Al-Rfou, R., Choe, D., Constant, N., Guo, M., & Jones, L. (2019). Character-level language modeling with deeper self-attention. *Proc. AAAI Conference on Artificial Intelligence*, *33*(01), 3159–3166 (cit. on p. 95).

Apple, W., Streeter, L. A., & Krauss, R. M. (1979). Effects of pitch and speech rate on personal attributions. *Journal of Personality and Social Psychology*, *37*(5), 715 (cit. on p. 106).

Bachorowski, J.-A. (1999). Vocal expression and perception of emotion. *Current directions in psychological science*, *8*(2), 53–57 (cit. on pp. 2, 21, 35, 138).

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, *33*, 12449–12460 (cit. on pp. 28, 30).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint* (cit. on pp. 11, 44).

Bailly, G., Berar, M., Elisei, F., & Odisio, M. (2003). Audiovisual speech synthesis. *International Journal of Speech Technology*, *6*, 331–346 (cit. on p. 144).

Bailly, G., & Gouvernayre, C. (2012). Pauses and respiratory markers of the structure of book reading. *Proc. Interspeech*, 2218–2221 (cit. on pp. 9, 64, 66, 80).

Barbosa, P., & Bailly, G. (1994). Characterisation of rhythmic patterns for text-to-speech synthesis. *Speech Communication*, *15*(1), 127–137 (cit. on p. 65).

Bau, A., Belinkov, Y., Sajjad, H., Durrani, N., Dalvi, F., & Glass, J. (2019). Identifying and controlling important neurons in neural machine translation. *Proc. ICLR* (cit. on pp. 28, 100).

Becker, K. (2014). Linguistic repertoire and ethnic identity in New York City. *Language & Communication*, *35*, 43–54 (cit. on pp. 1, 21).

Beckman, M. E., & Pierrehumbert, J. B. (1986). Intonational structure in Japanese and English. *Phonology*, *3*, 255–309 (cit. on p. 38).

Biocca, F., Burgoon, J., Harms, C., Stoner, M., et al. (2001). Criteria and scope conditions for a theory and measure of social presence. *Presence: Teleoperators and virtual environments*, *10*(01), 2001 (cit. on p. 2).

Black, A. W., Lenzo, K., & Pagel, V. (1998). Issues in building general letter to sound rules. *The third ESCA/COCOSDA workshop (ETRW) on speech synthesis* (cit. on p. 176).

Boersma, P. (2001). Praat, a system for doing phonetics by computer. *Glot. Int.*, *5*(9), 341–345 (cit. on pp. 66, 86).

Bolinger, D. (1989). *Intonation and its uses: Melody in grammar and discourse*. Stanford university press. (Cit. on pp. 1, 2, 23).

Bolton, R. N., McColl-Kennedy, J. R., Cheung, L., Gallan, A., Orsingher, C., Witell, L., & Zaki, M. (2018). Customer experience challenges: bringing together digital, physical and social realms. *Journal of service management*, *29*(5), 776–808 (cit. on p. 2).

Bosse, M.-L., & Valdois, S. (2009). Influence of the visual attention span on child reading performance: a cross-sectional study. *Journal of Research in Reading*, *32*(2), 230–253 (cit. on pp. 8, 11, 41, 42, 48).

Brognaux, S., & Avanzi, M. (2015). Sociophonetics of phonotactic phenomena in French. *ICPhS*, 5 pages (cit. on p. 7).

Brown, G., Currie, K. L., & Kenworthy, J. (1980). *Questions of intonation*. (Cit. on pp. 21, 23).

Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, *70*, 245–317 (cit. on p. 84).

Campione, E., & Véronis, J. (2002). A large-scale multilingual study of silent pause duration. *Speech Prosody* (cit. on p. 80).

Cassell, J. (2000). Embodied conversational interface agents. *Communications of the ACM*, *43*(4), 70–78 (cit. on p. 2).

Chen, L.-W., & Rudnicky, A. (2022). Fine-grained style control in Transformer-based Text-to-speech Synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7907–7911 (cit. on pp. 84, 126).

Chen, M., Tan, X., Ren, Y., Xu, J., Sun, H., Zhao, S., Qin, T., & Liu, T.-Y. (2020). Multispeech: Multi-speaker text to speech with transformer. *Proc. Interspeech* (cit. on p. 24).

Chien, C.-M., Lin, J.-H., Huang, C.-y., Hsu, P.-c., & Lee, H.-y. (2021). Investigating on Incorporating Pretrained and Learnable Speaker Representations for Multi-Speaker Multi-Style Text-to-Speech. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8588–8592 (cit. on pp. 18, 25).

Cho, C. J., Wu, P., Mohamed, A., & Anumanchipalli, G. K. (2023). Evidence of Vocal Tract Articulation in Self-Supervised Learning of Speech. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 30).

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in Neural Information Processing Systems*, 577–585 (cit. on pp. 11, 13, 40, 47).

Clark, R., Silen, H., Kenter, T., & Leith, R. (2019). Evaluating long-form text-to-speech: Comparing the ratings of sentences and paragraphs. *Proc. 10th ISCA Workshop on Speech Synthesis (SSW)*, 99–104 (cit. on p. 33).

Clifton, A., Reddy, S., Yu, Y., Pappu, A., Rezapour, R., Bonab, H., Eskevich, M., Jones, G., Karlgren, J., Carterette, B., et al. (2020). 100,000 podcasts: A spoken English document corpus. *Proc. 28th International Conference on Computational Linguistics*, 5903–5917 (cit. on p. 9).

Cole, J., & Shattuck-Hufnagel, S. (2016). New methods for prosodic transcription: Capturing variability as a source of information. *Laboratory Phonology*, *7*(1) (cit. on pp. xvii, 36, 120, 155).

Cox, M. A., & Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization* (pp. 315–347). Springer. (Cit. on p. 78).

Cruttenden, A. (1997). *Intonation*. Cambridge University Press. (Cit. on p. 21).

D'Agostino, R., & Pearson, E. S. (1973). Tests for departure from normality. Empirical results for the distributions of b^2 and root square b. *Biometrika*, *60*(3), 613–622 (cit. on p. 119).

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of Oz studies: why and how. *Proc. 1st International Conference on Intelligent User Interfaces*, 193–200 (cit. on p. 37).

Dahmani, S., Colotte, V., Girard, V., & Ouni, S. (2019). Conditional variational auto-encoder for text-driven expressive audiovisual speech synthesis. *Proc. Interspeech* (cit. on p. 144).

Dahmani, S., Colotte, V., Girard, V., & Ouni, S. (2021). Learning emotions latent representation with CVAE for text-driven expressive audiovisual speech synthesis. *Neural Networks*, *141*, 315–329 (cit. on p. 144).

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. *International conference on machine learning*, 933–941 (cit. on p. 18).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proc. NAACL-HLT* (cit. on pp. 59, 66, 126, 141).

Dunn, R. H., & Vitolins, V. (2019). eSpeak NG speech synthetizer. (Cit. on pp. 8, 50).

Ekman, P., et al. (1999). Basic emotions. *Handbook of cognition and emotion*, *98*(45-60), 16 (cit. on pp. 68, 127).

Eyben, F., Scherer, K. R., Schuller, B. W., Sundberg, J., André, E., Busso, C., Devillers, L. Y., Epps, J., Laukka, P., Narayanan, S. S., et al. (2015). The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE Transactions on Affective Computing*, *7*(2), 190–202 (cit. on p. 85).

Feinberg, D. R., Jones, B. C., Little, A. C., Burt, D. M., & Perrett, D. I. (2005). Manipulations of fundamental and formant frequencies influence the attractiveness of human male voices. *Animal behaviour*, *69*(3), 561–568 (cit. on p. 106).

Foulkes, P., & Docherty, G. (2006). The social life of phonetics and phonology. *Journal of Phonetics*, *34*(4), 409–438 (cit. on pp. 1, 21, 50).

Garnier, M., Ménard, L., & Richard, G. (2012). Effect of being seen on the production of visible speech cues. A pilot study on Lombard speech. *Proc. Interspeech*, 611–614 (cit. on p. 146).

Gobl, C., Bennett, E., & Chasaide, A. N. (2002). Expressive synthesis: how crucial is voice quality? *Proc. IEEE Workshop on Speech Synthesis*, 91–94 (cit. on p. 133).

Godde, E., Bailly, G., Escudero, D., Bosse, M.-L., & Gillet-Perret, E. (2017). Evaluation of reading performance of primary school children: Objective measurements vs. subjective ratings. *International workshop on child computer interaction (WOCCI)* (cit. on p. 125).

Govalkar, P., Fischer, J., Zalkow, F., & Dittmar, C. (2019). A comparison of recent neural vocoders for speech signal reconstruction. *Proc. 10th ISCA Speech Synthesis Workshop (SSW)*, 7–12 (cit. on pp. 19, 185).

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint* (cit. on p. 11).

Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *32*(2), 236–243 (cit. on p. 8).

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. *Proc. Interspeech*, 5036–5040 (cit. on pp. 17, 18).

Gunning, D. (2017). Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*, *2*(2), 1 (cit. on p. 84).

Guo, P., Boyer, F., Chang, X., Hayashi, T., Higuchi, Y., Inaguma, H., Kamo, N., Li, C., Garcia-Romero, D., Shi, J., et al. (2021). Recent developments on espnet toolkit boosted by conformer. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5874–5878 (cit. on p. 18).

Gutierrez, E., Oplustil-Gallegos, P., & Lai, C. (2021). Location, Location: Enhancing the Evaluation of Text-to-Speech synthesis using the Rapid Prosody Transcription Paradigm. *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 25–30 (cit. on pp. 35, 36, 120).

Hazan, V., & Baker, R. (2011). Acoustic-phonetic characteristics of speech produced with communicative intent to counter adverse listening conditions. *The Journal of the Acoustical Society of America*, *130*(4), 2139–2152 (cit. on p. 24).

He, M., Deng, Y., & He, L. (2019). Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS. *Proc. Interspeech* (cit. on pp. 14, 41, 42).

Hinterleitner, F., Möller, S., Norrenbrock, C., & Heute, U. (2011). Perceptual quality dimensions of text-to-speech systems. *Twelfth Annual Conference of the International Speech Communication Association* (cit. on p. 35).

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint* (cit. on p. 121).

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A field guide to dynamical recurrent neural networks*. IEEE Press. (Cit. on pp. 14, 40, 62).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780 (cit. on pp. 12–14, 40).

Honnet, P.-E., Lazaridis, A., Garner, P. N., & Yamagishi, J. (2017). *The Siwis French Speech Synthesis Database. Design and recording of a high quality french database for speech synthesis* (tech. rep.). Idiap. (Cit. on pp. 9, 54, 63, 173, 174).

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3451–3460 (cit. on pp. 28, 30).

Hsu, W.-N., Zhang, Y., Weiss, R. J., Chung, Y.-A., Wang, Y., Wu, Y., & Glass, J. (2019). Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5901–5905 (cit. on pp. 25, 26, 30).

Huang, R., Lam, M. W. Y., Wang, J., Su, D., Yu, D., Ren, Y., & Zhao, Z. (2022). FastDiff: A Fast Conditional Diffusion Model for High-Quality Speech Synthesis [Main Track]. In L. D. Raedt (Ed.), *Proc. 31st International Joint Conference on Artificial Intelligence, IJCAI-22* (pp. 4157–4163). (Cit. on pp. 20, 154).

Hussen Abdelaziz, A., Kumar, A. P., Seivwright, C., Fanelli, G., Binder, J., Stylianou, Y., & Kajareker, S. (2021). Audiovisual speech synthesis using Tacotron2. *Proc. 2021 International Conference on Multimodal Interaction*, 503–511 (cit. on pp. 14, 144, 148, 151).

International Telecommunication Union, ITU. (1998). *Methods for objective and subjective assessment of quality* (tech. rep. ITU-T P.800). International Telecommunication Union. (Cit. on p. 117).

International Telecommunications Union, I. (2003). 1534-1, Method for the subjective assessment of intermediate quality level of coding systems. *International Telecommunications Union, Geneva, Switzerland*, *14* (cit. on pp. 34, 70, 76, 136).

Ito, K., & Johnson, L. (2017). The LJ Speech Dataset. (Cit. on pp. 63, 64).

Jacquelin, M., Garnier, M., Girin, L., Vincent, R., & Perrotin, O. (2023). Exploring the multidimensional representation of individual speech acoustic parameters extracted by deep unsupervised models. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 240–241 (cit. on pp. 97, 121).

Jaunet, T., Kervadec, C., Vuillemot, R., Antipov, G., Baccouche, M., & Wolf, C. (2021). Visqa: X-raying vision and language reasoning in transformers. *IEEE Transactions on Visualization and Computer Graphics*, *28*(1), 976–986 (cit. on p. 47).

Ji, Y., Cohn, T., Kong, L., Dyer, C., & Eisenstein, J. (2016). Document context language models. *Proc. ICLR* (cit. on p. 62).

Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., Chen, Z., Nguyen, P., Pang, R., Moreno, I. L., et al. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Proc. NeurIPS* (cit. on pp. 25, 30, 84, 85).

Joly, A., Nicolis, M., Peterova, E., Lombardi, A., Abbas, A., van Korlaar, A., Hussain, A., Sharma, P., Moinet, A., Lajszczak, M., Karanasou, P., Bonafonte, A., Drugman, T., & Sokolova, E. (2023). Controllable Emphasis with Zero Data for Text-To-Speech. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 113–119 (cit. on p. 108).

Junqua, J.-C. (1996). The influence of acoustics on speech production: A noise-induced stress phenomenon known as the Lombard reflex. *Speech Communication*, *20*(1-2), 13–22 (cit. on p. 24).

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., Oord, A., Dieleman, S., & Kavukcuoglu, K. (2018). Efficient neural audio synthesis. *International Conference on Machine Learning*, 2410–2419 (cit. on pp. 19, 185).

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. *Annual Meeting of the Association for Computational Linguistics*, 655–665 (cit. on p. 12).

Kastner, K., Santos, J. F., Bengio, Y., & Courville, A. (2019). Representation mixing for TTS synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5906–5910 (cit. on pp. 38, 44).

Kawahara, H., Masuda-Katsuse, I., & de Cheveigné, A. (1999). Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, *27*(3-4), 187–207 (cit. on p. 20).

Kayyar, K., Dittmar, C., Pia, N., & Habets, E. (2023). Subjective Evaluation of Text-to-Speech Models: Comparing Absolute Category Rating and Ranking by Elimination Tests. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 191–196 (cit. on p. 34).

Kearns, J. (2014). LibriVox: Free public domain audiobooks. *Reference Reviews* (cit. on pp. 8, 54, 63, 64, 173).

Kenter, T., Wan, V., Chan, C.-A., Clark, R., & Vit, J. (2019). CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. *International Conference on Machine Learning*, 3331–3340 (cit. on pp. 10, 126).

Kim, J., Kong, J., & Son, J. (2021). Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. *International Conference on Machine Learning*, 5530–5540 (cit. on p. 19).

Kim, M., Cheon, S. J., Choi, B. J., Kim, J. J., & Kim, N. S. (2021). Expressive Text-to-Speech Using Style Tag. *Proc. Interspeech*, 4663–4667 (cit. on pp. 18, 30, 84, 140, 155).

Kim, T.-H., Cho, S., Choi, S., Park, S., & Lee, S.-Y. (2020). Emotional voice conversion using multitask learning with text-to-speech. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7774–7778 (cit. on p. 25).

King, S., & Karaiskos, V. (2013). The Blizzard Challenge 2013. *Proc. 9th Blizzard Challenge Workshop* (cit. on p. 35).

Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *proc. ICLR* (cit. on p. 19).

Kirkland, A., Mehta, S., Lameris, H., Henter, G. E., Szekely, E., & Gustafson, J. (2023). Stuck in the MOS pit: A critical analysis of MOS test methodology in TTS evaluation. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 41–47 (cit. on pp. 33, 34).

Klapsas, K., Ellinas, N., Sung, J. S., Park, H., & Raptis, S. (2021). Word-level style control for expressive, non-attentive speech synthesis. *SPECOM*, 336–347 (cit. on pp. 126, 134).

Klimkov, V., Ronanki, S., Rohnke, J., & Drugman, T. (2019). Fine-Grained Robust Prosody Transfer for Single-Speaker Neural Text-To-Speech. *Proc. Interspeech*, 4440–4444 (cit. on p. 14).

Knowles, G. (1987). Patterns of spoken English: an introduction to English Phonetics. (Cit. on p. 24).

Koch, J., Lux, F., Schauffler, N., Bernhart, T., Dieterle, F., Kuhn, J., Richter, S., Viehhauser, G., & Thang Vu, N. (2022). PoeticTTS - Controllable Poetry Reading for Literary Studies. *Proc. Interspeech*, 1223–1227 (cit. on p. 108).

Koizumi, Y., Yatabe, K., Zen, H., & Bacchiani, M. (2023). WaveFit: An iterative and non-autoregressive neural vocoder based on fixed-point iteration. *2022 IEEE Spoken Language Technology Workshop (SLT)*, 884–891 (cit. on p. 19).

Kong, J., Kim, J., & Bae, J. (2020). Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, *33*, 17022–17033 (cit. on pp. 19–21, 186, 187).

Krause, J. C., & Braida, L. D. (2004). Acoustic properties of naturally produced clear speech at normal speaking rates. *The Journal of the Acoustical Society of America*, *115*(1), 362–378 (cit. on p. 24).

Kruskal, J. B., & Wish, M. (1978). *Multidimensional scaling*. Sage. (Cit. on pp. 35, 71, 87).

Kubichek, R. (1993). Mel-cepstral distance measure for objective speech quality assessment. *Pacific Rim Conference on Communications Computers and Signal Processing*, *1*, 125–128 (cit. on pp. 69, 71, 78, 133).

Ladd, D. R. (2008). *Intonational phonology*. Cambridge University Press. (Cit. on p. 22).

Łańcucki, A. (2021). Fastpitch: Parallel text-to-speech with pitch prediction. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6588–6592 (cit. on p. 18).

Laranjo, L., Dunn, A. G., Tong, H. L., Kocaballi, A. B., Chen, J., Bashir, R., Surian, D., Gallego, B., Magrabi, F., Lau, A. Y., et al. (2018). Conversational agents in healthcare: a systematic review. *Journal of the American Medical Informatics Association*, *25*(9), 1248–1258 (cit. on p. 2).

Latif, S., Qadir, J., Qayyum, A., Usama, M., & Younis, S. (2020). Speech technology for healthcare: Opportunities, challenges, and state of the art. *IEEE Reviews in Biomedical Engineering*, *14*, 342–356 (cit. on p. 1).

Latorre, J., Yanagisawa, K., Wan, V., Kolluru, B., & Gales, M. J. (2014). Speech intonation for TTS: Study on evaluation methodology. *Fifteenth Annual Conference of the International Speech Communication Association* (cit. on pp. 33, 34, 75).

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., & Schwab, D. (2020). FlauBERT: des modèles de langue contextualisés pré-entrainés pour le français (FlauBERT: Unsupervised Language Model Pre-training for French). *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2: Traitement Automatique des Langues Naturelles*, 268–278 (cit. on p. 140).

Le Moine, C., & Obin, N. (2020). Att-HACK: An Expressive Speech Database with Social Attitudes. *Speech Prosody* (cit. on pp. 9, 23, 127).

Lee, S.-g., Ping, W., Ginsburg, B., Catanzaro, B., & Yoon, S. (2023). BigVGAN: A Universal Neural Vocoder with Large-Scale Training. *The Eleventh International Conference on Learning Representations* (cit. on p. 20).

Lee, Y., Rabiee, A., & Lee, S.-Y. (2017). Emotional End-to-End Neural Speech Synthesizer (cit. on p. 25).

Lei, S., Zhou, Y., Chen, L., Wu, Z., Kang, S., & Meng, H. (2022). Towards expressive speaking style modelling with hierarchical context information for mandarin speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7922–7926 (cit. on p. 18).

Lei, Y., Yang, S., Wang, X., & Xie, L. (2022). Msemotts: Multi-scale emotion transfer, prediction, and control for emotional speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *30*, 853–864 (cit. on p. 126).

Li, N., Liu, S., Liu, Y., Zhao, S., & Liu, M. (2019). Neural speech synthesis with transformer network. *Proc. AAAI Conference on Artificial Intelligence*, *33*(01), 6706–6713 (cit. on pp. 15, 16, 47).

Liberman, M., & Prince, A. (1977). On stress and linguistic rhythm. *Linguistic inquiry*, *8*(2), 249–336 (cit. on pp. 62, 124).

Liénard, J.-S., & Di Benedetto, M.-G. (1999). Effect of vocal effort on spectral properties of vowels. *The Journal of the Acoustical Society of America*, *106*(1), 411–422 (cit. on p. 110).

Lim, D., Jung, S., & Kim, E. (2022). JETS: Jointly Training FastSpeech2 and HiFi-GAN for End to End Text to Speech. *Proc. Interspeech*, 21–25 (cit. on p. 21).

Liu, B., & Lane, I. (2017). Dialog context language modeling with recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5715–5719 (cit. on p. 62).

Liu, R., Sisman, B., Gao, G., & Li, H. (2021). Expressive TTS training with frame and style reconstruction loss. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 1806–1818 (cit. on p. 14).

Maeda, S. (1976). A characterization of American English intonation (cit. on p. 22).

Maier, J. X., Di Luca, M., & Noppeney, U. (2011). Audiovisual asynchrony detection in human speech. *Journal of Experimental Psychology: Human Perception and Performance*, *37*(1), 245 (cit. on p. 144).

Mattheyses, W., & Verhelst, W. (2015). Audiovisual speech synthesis: An overview of the state-of-the-art. *Speech Communication*, *66*, 182–217 (cit. on p. 144).

Mayo, C., Clark, R. A., & King, S. (2005). Multidimensional scaling of listener responses to synthetic speech (cit. on p. 35).

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). Montreal forced aligner: Trainable text-speech alignment using kaldi. *Proc. Interspeech*, *2017*, 498–502 (cit. on p. 11).

McFarland, D. H. (2001). Respiratory markers of conversational interaction (cit. on p. 9).

McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, *264*(5588), 746–748 (cit. on p. 146).

McNeill, D. (2019). *Gesture and thought*. University of Chicago press. (Cit. on p. 143).

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., & Bengio, Y. (2016). SampleRNN: An unconditional end-to-end neural audio generation model. *proc. ICLR* (cit. on p. 10).

Mehta, S., Wang, S., Alexanderson, S., Beskow, J., Szekely, E., & Henter, G. E. (2023). Diff-TTSG: Denoising probabilistic integrated speech and gesture synthesis. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 150–156 (cit. on pp. 144, 154).

Miao, C., Shuang, L., Liu, Z., Minchuan, C., Ma, J., Wang, S., & Xiao, J. (2021). Efficienttts: An efficient and high-quality text-to-speech architecture. *International Conference on Machine Learning*, 7700–7709 (cit. on p. 21).

Min, D., Lee, D. B., Yang, E., & Hwang, S. J. (2021). Meta-stylespeech: Multi-speaker adaptive text-to-speech generation. *International Conference on Machine Learning*, 7748–7759 (cit. on pp. 2, 18).

Modarresi, G., Sussman, H., Lindblom, B., & Burlingame, E. (2004). An acoustic analysis of the bidirectionality of coarticulation in VCV utterances. *Journal of Phonetics*, *32*(3), 291–312 (cit. on p. 96).

Mohan, D. S. R., Hu, V., Teh, T. H., Torresquintero, A., Wallis, C. G., Staib, M., Foglianti, L., Gao, J., & King, S. (2021). Ctrl-P: Temporal Control of Prosodic Variation for Speech Synthesis. *Proc. Interspeech*, 3875–3879 (cit. on pp. 14, 85, 108).

Morise, M., Kawahara, H., & Katayose, H. (2009). Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. *Audio Engineering Society Conference: 35th International Conference: Audio for Games* (cit. on p. 51).

Mustafa, A., Pia, N., & Fuchs, G. (2021). Stylemelgan: An efficient high-fidelity adversarial vocoder with temporal adaptive normalization. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6034–6038 (cit. on p. 20).

Nenkova, A., Brenier, J., Kothari, A., Calhoun, S., Whitton, L., Beaver, D., & Jurafsky, D. (2007). To Memorize or to Predict: Prominence labeling in Conversational Speech. In Sidner, Candace and Schultz, Tanja and Stone, Matthew and Zhai, ChengXiang (Ed.), *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proc. Main Conference* (pp. 9–16). Association for Computational Linguistics. (Cit. on p. 7).

Nick Campbell. (1992). *Multi-level Timing in Speech* (Doctoral dissertation). Sussex University, U.K. Department of Experimental Psychology. (Cit. on pp. 10, 108, 112).

Novitasari, S., Sakti, S., & Nakamura, S. (2022). A machine speech chain approach for dynamically adaptive Lombard TTS in static and dynamic noise environments. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *30*, 2673–2688 (cit. on p. 24).

Nyatsanga, S., Kucherenko, T., Ahuja, C., Henter, G. E., & Neff, M. (2023). A Comprehensive Review of Data-Driven Co-Speech Gesture Generation. *Computer Graphics Forum*, *42*(2), 569–596 (cit. on p. 143).

O'Mahony, J., Lai, C., & King, S. (2022). Combining conversational speech with read speech to improve prosody in Text-To-Speech synthesis. In *Proc. Interspeech*. (Cit. on p. 9).

O'Mahony, J., Oplustil-Gallegos, P., Lai, C., & King, S. (2021). Factors Affecting the Evaluation of Synthetic Speech in Context. *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 148–153 (cit. on p. 34).

Oplustil-Gallegos, P., O'Mahony, J., & King, S. (2021). Comparing acoustic and textual representations of previous linguistic context for improving Text-to-Speech. *Proc. 11th ISCA Speech Synthesis Workshop (SSW)*, 205–210 (cit. on pp. 62, 66).

Osborne, J. W., & Overbay, A. (2004). The power of outliers (and why researchers should always check for them). *Practical Assessment, Research, and Evaluation*, *9*(1), 6 (cit. on p. 35).

O'Shaughnessy, D. (1981). A study of French vowel and consonant durations. *Journal of Phonetics*, *9*(4), 385–406 (cit. on pp. 10, 44).

Palan, S., & Schitter, C. (2018). Prolific. ac—A subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, *17*, 22–27 (cit. on pp. 70, 77, 118, 148).

Pascual, S., Serrà, J., & Bonafonte, A. (2019). Exploring efficient neural architectures for linguistic–acoustic mapping in text-to-speech. *Applied Sciences*, *9*(16), 3391 (cit. on p. 62).

Perquin, A., Cooper, E., & Yamagishi, J. (2020). An Investigation of the Relation Between Grapheme Embeddings and Pronunciation for Tacotron-based Systems. *arXiv preprint* (cit. on pp. 29–31, 48, 56, 94).

Perrotin, O. (2021). *An Introduction to Neural Vocoders* (tech. rep.). GIPSA-lab. (Cit. on p. 20).

Perrotin, O., Amouri, H., Bailly, G., & Hueber, T. (2021). Evaluating the Extrapolation Capabilities of Neural Vocoders to Extreme Pitch Values. *Proc. Interspeech*, 11–15 (cit. on pp. 20, 140).

Perrotin, O., Stpehenson, B., Gerber, S., & Bailly, G. (2023). The Blizzard Challenge 2023. *Proc. 18th Blizzard Challenge Workshop*, 1–27 (cit. on pp. 18, 20, 32, 35, 36, 50, 53, 59, 63, 174).

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., & Kudinov, M. (2021). Grad-tts: A diffusion probabilistic model for text-to-speech. *International Conference on Machine Learning*, 8599–8608 (cit. on p. 154).

Potdevin, D. (2020). *Vers des agents conversationnels animés sociaux: Quelle influence de l'intimité virtuelle sur l'expérience utilisateur et la relation-client?* (Doctoral dissertation). Université Paris-Saclay. (Cit. on pp. 1, 2, 23, 143).

Prechelt, L. (2002). Early stopping-but when? In *Neural Networks: Tricks of the trade* (pp. 55–69). Springer. (Cit. on p. 33).

Prenger, R., Valle, R., & Catanzaro, B. (2019). Waveglow: A flow-based generative network for speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3617–3621 (cit. on pp. 19, 20, 186).

Puts, D. A., Gaulin, S. J., & Verdolini, K. (2006). Dominance and the evolution of sexual dimorphism in human voice pitch. *Evolution and human behavior*, *27*(4), 283–296 (cit. on p. 106).

Raitio, T., Rasipuram, R., & Castellani, D. (2020). Controllable Neural Text-to-Speech Synthesis Using Intuitive Prosodic Features. *Proc. Interspeech*, 4432–4436 (cit. on pp. 14, 85, 108).

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T.-Y. (2021). FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. *International Conference on Learning Representations* (cit. on pp. 7, 10, 12, 16, 21, 33, 38, 40, 85, 96, 126, 128, 134, 144, 145, 181).

Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., & Liu, T.-Y. (2019). Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 3171–3180 (cit. on pp. xvii, 11, 16, 17).

Resnick, M. C. (2012). *Phonological variants and dialect identification in Latin American Spanish* (Vol. 201). Walter de Gruyter. (Cit. on pp. 1, 21, 50).

Sainburg, T. (2019). *timsainb/noisereduce: v1.0* (Version db94fe2). Zenodo. (Cit. on p. 187).

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised Pre-Training for Speech Recognition. *Proc. Interspeech*, 3465–3469 (cit. on p. 28).

Schoeffler, M., Bartoschek, S., Stöter, F.-R., Roess, M., Westphal, S., Edler, B., & Herre, J. (2018). webMUSHRA—A comprehensive framework for web-based listening tests. *Journal of Open Research Software*, *6*(1) (cit. on pp. 34, 76, 136).

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, *45*(11), 2673–2681 (cit. on pp. 12, 14).

Selkirk, E. (1986). On derived domains in sentence phonology. *Phonology*, *3*, 371–405 (cit. on pp. 38, 124).

Shen, J., Jia, Y., Chrzanowski, M., Zhang, Y., Elias, I., Zen, H., & Wu, Y. (2020). Non-attentive tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling. *arXiv preprint* (cit. on pp. 10, 14, 40, 41, 62, 125).

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., et al. (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4779–4783 (cit. on pp. 7, 10, 12, 13, 32, 33, 38, 40, 44, 67, 126, 179).

Shin, Y., Lee, Y., Jo, S., Hwang, Y., & Kim, T. (2022). Text-driven Emotional Style Control and Cross-speaker Style Transfer in Neural TTS. *Proc. Interspeech*, 2313–2317 (cit. on pp. 30, 84, 140, 155).

Shirahata, Y., Yamamoto, R., Song, E., Terashima, R., Kim, J.-M., & Tachibana, K. (2023). Period VITS: Variational Inference with Explicit Pitch Modeling for End-To-End Emotional Speech Synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (cit. on p. 19).

Shirali-Shahreza, S., & Penn, G. (2023). Better Replacement for TTS Naturalness Evaluation. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 197–203 (cit. on p. 33).

Sigurgeirsson, A. T., & King, S. (2023). Using a Large Language Model to Control Speaking Style for Expressive TTS. *Proc. 12th ISCA Speech Synthesis Workshop (SSW)*, 246–247 (cit. on p. 108).

Sini, A., Lolive, D., Vidal, G., Tahon, M., & Delais-Roussarie, É. (2018). Synpaflex-corpus: An expressive french audiobooks corpus dedicated to expressive speech synthesis. *Proc. Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (cit. on pp. 8, 9, 68, 80, 127).

Skantze, G. (2021). Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, *67*, 101178 (cit. on p. 36).

Skerry-Ryan, R., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R. J., Clark, R., & Saurous, R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *Proc. ICML* (cit. on pp. 2, 14, 24–26, 30, 85, 124, 128).

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 5329–5333 (cit. on p. 25).

Solak, I. (2019). The M-AILABS speech dataset. (Cit. on pp. 8, 9, 55, 63, 68).

Sorin, A., Shechtman, S., & Hoory, R. (2020). Principal Style Components: Expressive Style Control and Cross-Speaker Transfer in Neural TTS. *Proc. Interspeech*, 3411–3415 (cit. on pp. 27, 30).

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., & Bengio, Y. (2017). Char2wav: End-to-end speech synthesis. *proc. ICLR* (cit. on p. 9).

Stanton, D., Wang, Y., & Skerry-Ryan, R. (2018). Predicting expressive speaking style from text in end-to-end speech synthesis. *2018 IEEE Spoken Language Technology Workshop (SLT)*, 595–602 (cit. on p. 126).

Stephenson, B., Besacier, L., Girin, L., & Hueber, T. (2020). What the Future Brings: Investigating the Impact of Lookahead for Incremental Neural TTS. *Proc. Interspeech*, 215–219 (cit. on p. 12).

Stephenson, B., Besacier, L., Girin, L., & Hueber, T. (2022). BERT, can HE predict contrastive focus? Predicting and controlling prominence in neural TTS using a language model. *Proc. Interspeech*, 3383–3387 (cit. on pp. 7, 66).

Stevens, S. S., & Volkmann, J. (1940). The relation of pitch to frequency: A revised scale. *The American Journal of Psychology*, *53*(3), 329–353 (cit. on p. 8).

Streeck, J., & Knapp, M. L. (1992). The interaction of visual and verbal features in human communication. *Advances in Nonverbal Communication*, *10*, 3–23 (cit. on p. 1).

Stuart-Smith, J., Lawson, E., & Scobbie, J. M. (2014). Derhoticisation in Scottish English. *Advances in Sociophonetics*, *15*, 59 (cit. on pp. 1, 7, 21, 50).

Stylianou, Y. (2001). Applying the harmonic plus noise model in concatenative speech synthesis. *IEEE Transactions on speech and audio processing*, *9*(1), 21–29 (cit. on p. 20).

Székely, É., Henter, G. E., Beskow, J., & Gustafson, J. (2019). Spontaneous Conversational Speech Synthesis from Found Data. *Proc. Interspeech*, 4435–4439 (cit. on p. 9).

Tahon, M., Qader, R., Lecorvé, G., & Lolive, D. (2016). Improving TTS with Corpus-Specific Pronunciation Adaptation. *Proc. Interspeech*, 2831–2835 (cit. on p. 50).

Tan, X. (2023). Vocoders. In *Neural text-to-speech synthesis* (pp. 101–114). Springer Nature Singapore. (Cit. on p. 19).

Taylor, J., & Richmond, K. (2019). Analysis of Pronunciation Learning in End-to-End Speech Synthesis. *Proc. Interspeech*, 2070–2074 (cit. on pp. 8, 50, 56, 58).

Taylor, P. (2009). Text-to-speech synthesis. Cambridge university press. (Cit. on p. 6).

Theobald, B. (2007). Audiovisual speech synthesis. *International Congress on Phonetic Sciences*, 285–290 (cit. on p. 144).

Tits, N., El Haddad, K., & Dutoit, T. (2021). Analysis and assessment of controllability of an expressive deep learning-based tts system. *Informatics*, *8 (4)*, 84 (cit. on pp. 30, 85, 88, 100).

Tits, N., Wang, F., Haddad, K. E., Pagel, V., & Dutoit, T. (2019). Visualization and Interpretation of Latent Spaces for Controlling Expressive Speech Synthesis through Audio Analysis. *Proc. Interspeech* (cit. on pp. 30, 31, 85, 100).

Triantafyllopoulos, A., Schuller, B. W., İymen, G., Sezgin, M., He, X., Yang, Z., Tzirakis, P., Liu, S., Mertes, S., André, E., et al. (2023). An overview of affective speech synthesis and conversion in the deep learning era. *Proc. IEEE* (cit. on p. 13).

Vaidya, A. R., Jain, S., & Huth, A. G. (2022). Self-supervised models of audio effectively explain human cortical responses to speech. *Proc. ICML* (cit. on pp. 28, 29, 97).

Vainio, M., Suni, A., & Aalto, D. (2013). Continuous wavelet transform for analysis of speech prosody. *Tools and Resources for the Analysys of Speech Prosody (TRASP), Laboratoire Parole et Language, Aix-en-Provence, France* (cit. on pp. xvii, 17, 51, 126).

Vaissière, J. (2002). Cross-linguistic prosodic transcription: French vs. English. (Cit. on p. 22).

Valin, J.-M., & Skoglund, J. (2019). LPCNet: Improving neural speech synthesis through linear prediction. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5891–5895 (cit. on p. 19).

Valle, R., Li, J., Prenger, R., & Catanzaro, B. (2020). Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6189–6193 (cit. on p. 14).

Van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio, 125 (cit. on pp. 6, 12, 19, 185).

Van Engen, K. J., Baese-Berk, M., Baker, R. E., Choi, A., Kim, M., & Bradlow, A. R. (2010). The Wildcat Corpus of native-and foreign-accented English: Communicative efficiency across conversational dyads with varying language alignment profiles. *Language and speech*, *53*(4), 510–540 (cit. on pp. 36, 37).

van Heuven, V. J., & van Bezooijen, R. (1995). Quality evaluation of synthesized speech. In *Speech coding and synthesis* (p. 707738). Elsevier Amsterdam. (Cit. on p. 33).

van Rijn, P., Mertes, S., Schiller, D., Harrison, P. M., Larrouy-Maestri, P., André, E., & Jacoby, N. (2021). Exploring Emotional Prototypes in a High Dimensional TTS Latent Space. *Proc. Interspeech*, 3870–3874 (cit. on p. 27).

Variani, E., Lei, X., McDermott, E., Moreno, I. L., & Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4052–4056 (cit. on p. 25).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30* (cit. on pp. 11, 15, 40, 62, 86, 95).

Veaux, C., Maia, R., & Papendreou, S. (2023). The DeepZen Speech Synthesis System for Blizzard Challenge 2023. *Proc. 18th Blizzard Challenge Workshop*, 81–86 (cit. on pp. 126, 131).

Wagner, P., Beskow, J., Betz, S., Edlund, J., Gustafson, J., Eje Henter, G., Le Maguer, S., Malisz, Z., Székely, É., Tånnander, C., et al. (2019). Speech synthesis evaluation—state-of-the-art assessment and suggestion for a novel research program. *Proc. 10th Speech Synthesis Workshop (SSW)* (cit. on p. 36).

Wang, S., Qian, Y., & Yu, K. (2017). What does the speaker embedding encode? *Proc. Interspeech*, 1497–1501 (cit. on pp. 30, 85).

Wang, S., Alexanderson, S., Gustafson, J., Beskow, J., Henter, G. E., & Székely, É. (2021). Integrated speech and gesture synthesis. *Proc. 2021 International Conference on Multimodal Interaction*, 177–185 (cit. on pp. 144, 145).

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., & Saurous, R. A. (2017). Tacotron: Towards End-to-End Speech Synthesis. *Proc. Interspeech*, 4006–4010 (cit. on pp. 9, 48, 56).

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., & Saurous, R. A. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *Proc. ICML* (cit. on pp. xvii, 14, 26, 27, 30, 62, 100, 102, 122, 124, 126, 128, 129).

Wells, D., Tang, H., & Richmond, K. (2022). Phonetic Analysis of Self-supervised Representations of English Speech. *Proc. Interspeech*, 3583–3587 (cit. on p. 28).

Wichmann, A. (2000). The attitudinal effects of prosody, and how they relate to emotion. *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion* (cit. on pp. 21, 23).

Wiedemann, G., Remus, S., Chawla, A., & Biemann, C. (2019). Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *Proc. 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, 161–170 (cit. on p. 141).

Winkworth, A. L., Davis, P. J., Ellis, E., & Adams, R. D. (1994). Variability and consistency in speech breathing during reading: Lung volumes, speech intensity, and linguistic factors. *Journal of Speech, Language, and Hearing Research*, *37*(3), 535–556 (cit. on p. 65).

Wright, S., Nocedal, J., et al. (1999). Numerical optimization. *Springer Science*, *35*(67-68), 7 (cit. on p. 88).

Wu, P., Ling, Z., Liu, L., Jiang, Y., Wu, H., & Dai, L. (2019). End-to-end emotional speech synthesis using style tokens and semi-supervised training. *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 623–627 (cit. on pp. 27, 30, 100, 102, 122, 125, 128, 130).

Xu, Z., Zhang, S., Wang, X., Zhang, J., Wei, W., He, L., & Zhao, S. (2023). MuLanTTS The Microsoft Speech Synthesis System for Blizzard Challenge 2023. *Proc. 18th Blizzard Challenge Workshop*, 46–51 (cit. on pp. 18, 74).

Yamamoto, R., Song, E., & Kim, J.-M. (2020). Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6199–6203 (cit. on p. 19).

Yang, C., Shen, Y., & Zhou, B. (2021). Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision*, *129*(5), 1451–1466 (cit. on pp. 28, 29, 100).

Yolchuyeva, S., Németh, G., & Gyires-Tóth, B. (2019). Transformer Based Grapheme-to-Phoneme Conversion. *Proc. Interspeech*, 2095–2099 (cit. on p. 8).

Yoon, E., Yoon, H. S., Gowda, D., Eom, S., Kim, D., Harvill, J., Gao, H., Hasegawa-Johnson, M., Kim, C., & Yoo, C. D. (2023). Mitigating the Exposure Bias in Sentence-Level Grapheme-to-Phoneme (G2P) Transduction. *Proc. Interspeech*, 2028–2032 (cit. on p. 56).

Yu, C., Lu, H., Hu, N., Yu, M., Weng, C., Xu, K., Liu, P., Tuo, D., Kang, S., Lei, G., Su, D., & Yu, D. (2020). DurIAN: Duration Informed Attention Network for Speech Synthesis. *Proc. Interspeech*, 2027–2031 (cit. on pp. 11, 144).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, 818–833 (cit. on pp. 2, 28, 60, 84).

Zen, H., Agiomyrgiannakis, Y., Egberts, N., Henderson, F., & Szczepaniak, P. (2016). Fast, Compact, and High Quality LSTM-RNN Based Statistical Parametric Speech Synthesizers for Mobile Devices. *Proc. Interspeech*, 2273–2277 (cit. on p. 43).

Zhang, J.-X., Richmond, K., Ling, Z.-H., & Dai, L. (2021). TaLNet: Voice Reconstruction from Tongue and Lip Articulation with Transfer Learning from Text-to-Speech Synthesis. *35*(16), 14402–14410 (cit. on p. 1).

Zhang, Y.-J., Pan, S., He, L., & Ling, Z.-H. (2019). Learning latent representations for style control and transfer in end-to-end speech synthesis. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6945–6949 (cit. on pp. 14, 27, 30, 62, 85, 100, 102, 108, 122, 124).

Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Chen, Z., Skerry-Ryan, R., Jia, Y., Rosenberg, A., & Ramabhadran, B. (2019). Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning. *Proc. Interspeech*, 2080–2084 (cit. on p. 24).

Zhou, X., Ling, Z.-H., & King, S. (2020). The Blizzard Challenge 2020. *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 1–18 (cit. on p. 63).

Appendix A

Corpus Description

This appendix presents the training corpus used to train neural TTS systems for the studies presented in this manuscript. All resources are briefly presented in Section A.1. The recording of the Theradia dataset is further described in Section A.2. All text annotations and the phonetic alphabet used is described in Section A.3.

A.1 Resources Content

This corpus is the concatenation of various sources summarized in Table A.1. Two speakers are taken from LibriVox [Kearns, 2014]. LibriVox is a collection of public domain audiobooks read by various speakers in multiple languages, including French, English, German, Italian, Spanish and more. The quality of recording varies a lot between speakers: NEB was chosen for her good recording quality and the amount of available data. DG was later added to the corpus to include a Male Speaker, despite the overall poorer quality of his recordings. SIWIS [Honnet et al., 2017] is an expressive dataset recorded by a female voice talent RO. It includes audiobooks and French parliament debates, read with a focus on emphasis. Finally, audiovisual expressive recordings were added to this corpus as part of the Theradia project [Tarpin-Bernard et al., 2021] (see Section A.2 for further details).

Table A.1: Multi-Speaker Audio-Visual Dataset used in this manuscript. Durations are given in hh:mm:ss. All textual contents are phonetically aligned. Alignments with audio have been hand-checked. **AB: Audiobooks**, **PS: Parliament Sessions**, **HG: Homographs**. LibriVox: [Kearns, 2014], SIWIS: [Honnet et al., 2017], Theradia: [Tarpin-Bernard et al., 2021].

Speaker	**Metadata**		**Datatype**			**Quantity**		**Content**
	Dataset	**Gender**	**Audio**	**Visual**	**Expressive**[1]	**Duration**	**# Utt**	
NEB	LibriVox	Female	✓	✗	✗	33:33:41	44 029	AB
DG	LibriVox	Male	✓	✗	✗	6:17:22	7 539	AB
RO	SIWIS	Female	✓	✗	✓	0:35:21	586	AB + PS
IZ	Theradia	Female	✓	✓	✗	0:30:46	733	AB + PS
AD	Theradia	Female	✓	✓	✓	10:31:29	12 462	AB + PS
Dictionary	Robert	✗	✗	✗	✗	✗	95 879	Isolated Words
HG	Various Hajj et al., 2022	✗	✗	✗	✗	✗	17 285	HG in Context
Total	-	-	-	-	-	**51:28:39**	**178 513**	-

[1]We consider dataset as expressive when explicit expressive instructions were given to the speaker during recording sessions.

In addition, two text-only resources are added to this corpus. These textual content is extracted for various online resources described by Hajj et al., 2022, as well as from the online Robert dictionary. This text-only part of the dataset is used to train the TTS models on phonetic prediction from orthographic inputs. This phonetic prediction task is further described in Section 2.3.2.

Audio sequences were segmented into utterances following the segmentation procedure described in Section 3.1.1. The sampling rate is 22.05 kHz. All orthographic and phonetic transcriptions are augmented with the initial linking punctuation mark as described in Section 3.1.2.

A.2 Recording of the Theradia Dataset

A.2.1 Expressive data from exercise-in-style sessions

Our internal French dataset has been uttered by a French professional theater actress, referred as AD in the following[2]. Sentences are taken from the SIWIS database [Honnet et al., 2017], which is composed of isolated extracts from French Novels and French parliament debates. We first asked the speaker to utter these extracts without a specific expressive style. We refer to these initial recordings as "Narrative".

For expressive speech recording, 12 attitudes were selected, summed up in table A.2. The speaker was asked to utter the given sentences with the specified style during exercise-in-style sessions. During these sessions, the actress was prompted to start her utterance with a context sentence relative to the style being produced: "I am begging you" for "Pleading", "I do not believe it" for "Skeptical", "Really?" for "Surprised", etc. This context sentence was cut off the final recording. The content is uncorrelated from the expressed style, and sentences differ between styles. This dataset was recorded with an audio-visual setup at GIPSA-lab, as part of the Theradia project [Tarpin-Bernard et al., 2021]. As a reminder, the Theradia project aims at designing a virtual agent to accompany patients during online cognitive therapies. The set of styles was decided in collaboration with speech therapists, who expressed their expectations towards this virtual agent.

This setup allowed us to gather about 30 hours of raw audiovisual recordings. At the time of writing of this manuscript, only 10 hours have been trimmed and phonetically aligned. Durations available by style at the moment of training of our expressive TTS model are given in table A.2. Only the audio was used for the expressive control discussed in Chapter 6, but video recordings were also used to train an audiovisual expressive TTS, described in Chapter 7. Two hours of the "Narrative" portion of this corpus were shared for the Spoke Task of the Blizzard Challenge 2023 [Perrotin et al., 2023].

[2]Early recordings were performed by a postdoctoral female researcher referred as IZ in Table A.1. This setup was quickly abandoned due to the poor audio quality of recordings.

Table A.2: Expressive Dataset Recorded for Theradia [Tarpin-Bernard et al., 2021]. Durations are given in hh:mm:ss.

Style		Annotated	
English	**French**	**Duration**	**# Utt**
Angry	Colère	00:25:42	555
Comforting	Réconfortant·e	00:33:54	515
Committed	Déterminé·e	00:22:30	459
Enthusiastic	Enthousiaste	00:30:54	597
Obvious	Evidence	00:28:30	519
Playful	Espiègle	00:20:36	493
Pleading	Suppliant·e	00:36:06	636
Skeptical	Incrédule	00:31:24	652
Sorry	Désolé·e	00:25:18	471
Surprised	Surpris·e	00:28:24	535
Thoughtful	Pensif·ive	00:45:30	477
Narrative	Narratif	05:02:12	6542
	Total	**10:31:00**	**12461**

A.2.2 Visual Recordings

The expressive dataset described in Section A.2.1 was recorded with audio-visual settings. The visual features are used to animate a virtual agent for the Theradia project [Tarpin-Bernard et al., 2021]. The face of the speakers was recorded with a a Logitech StreamCam. The video frame-rate is 60 Hz. The speaker's facial animation parameters are then tracked by an external service provider[3]. These facial animations are used to create a profile by speaker, which emulates the animation of the virtual agent by morphing the tracked facial features into deformations of the 3D model of the avatar. 152 visual control features are computed to animate the avatar, also sampled at 60 Hz. Fig. A.1 illustrates the animation of the avatar by the tracking of visual features on the video.

Since some of these control features co-vary in time, we found easier to reduce the number of coefficients to a minimum set of independent visual features. We used Principal Component Analysis (PCA) to compute a set of 37 features referred to as **Action Units (AU)** from the original 152 facial features. These 37 AU are distributed among the principal facial landmarks: 6 for the position of the head, 6 for the eyes, 3 for the eyelids, 4 for the eyebrows, 5 for the jaw, 10 for the lips and 3 for the nose. These 37 AU are the visual features representations we use to train our visual decoder, in the same way as we use 80 mel-spectrogram coefficients for the audio decoder.

Similarly as audio features for which a vocoder is needed to convert the speech representation back to audible waveforms, the limited set of 37 AU should be converted back to the

[3]DynamicXYZ© performed the tracking with the software Grabber.

initial 153 facial features. Because the PCA performs a linear reduction of the initial space, the inverse manipulation is straightforward. The PCA computes the eigen-decomposition of the covariance matrix of the initial facial features: the set of eigenvectors V associated to the higher eigenvalues (higher proportion of variance explained) can be used as the transition matrix between the initial and the reduced space. Inversely, predicted AU features can be projected back into facial features by dot-product with the transpose of V: V^T.

Figure A.1: Animation of the virtual agent by the tracking of visual features on the video recording.

A.3 Text Annotations and Phonetic

The textual content was provided by the Gutenberg Project[4] for LibriVox recordings, and was provided by the original resources otherwise. End of paragraphs marks "§" were added before each carriage return, in combination with existing punctuation marks. Text is in UTF8. '«»', '¬', '~', '"' ,'()' , '[]' are respectively used for speaking quotes, turn switches, three dots, quoted expression, aside quotes, notes. 'ö' has been transcribed as 'oe' because of rare occurrences. An emphasis marker '#' is added around words or groups of words that are given a particular prominence by the speaker[5]. The sequential nature of the corpus is introduced by reporting the last punctuation mark of each utterance at the beginning of the following one. This text-augmentation process, called *linking punctuation*, is discussed in section 3.1.2.

The entire corpus was phonetized and aligned with the audio automatically using an external Letter-To-Sound (L2S) front-end [Black et al., 1998]. The alignment was then hand-

[4]https://www.gutenberg.org/

[5]Emphasis annotation has been performed by Gérard Bailly, director of this PhD thesis.

checked by Gérard Bailly to account for sociophonetic variants produced by the speaker. This speaker adaptation step includes: 1) the annotation of released liaisons (specific linking between the ending "r", "t", "n" or "s" of a word and the first vowel of the next word). Note that no distinctions are made between mandatory, optional or forbidden liaisons, 2) The specifications of optional vowel harmony choices[6], and 3) the annotation of schwas /ə/ in opposition to front rounded mid-vowels /oe/ when produced by the speaker. Note that phonetic transcription is performed by word, which enables to keep word boundaries and punctuation marks when using phonetic input. Although this hand-checking ensures the quality of the alignment, it is also time consuming, which explains the limited number of utterances extracted from each corpus.

The phonetic alignment uses 36 symbols given in table A.3 (consonants) and table A.4 (vowels). 2 additional phonetic symbols are added to this set: /_/ and /__/, to indicate muted phones (phonetic mapping of characters that are not pronounced) and silences (pauses in speech) respectively. Phonetic transcriptions are given by word using curly brackets '{}'. This transcription preserves word boundaries and punctuation even when using phonetic inputs. Also, this enables the combination of orthographic and phonetic transcriptions in the same input sequence. This combination was used to phonetically transcribe mispronounced words and proper names.

The same phonetic symbols are used as outputs of the phonetic prediction layer described in section 2.3.2. In the proposed one-to-one L2S alignment, most letters produce only one phone (or none in the case of muted characters). However, some alignments map one letter to up to 3 phones. In this case, output phones are combined into diphones or triphones using the symbol '&'. For example, "expatrier" is aligned with "e^ **k&s** p a t r **i&j** e _".

[6]Vowel Harmony is the optional adaptations of vowels within a word in order for all vowels to share certain phonological features: frontness or backness, rounding, nasality... Example: "J'ôte" [o] VS "Nous ôtons" [ɔ]

Table A.3: French Consonants annotated in the corpus.

Articulation Point	Fricative		Plosive		Nasal	Lateral
	Voiceless	Voiced	Voiceless	Voiced		
Bilabial			p	b	m	
Labiodental	f	v				
Alveolar	s	z	t	d	n	l
Palatal	s^	z^			n~	
Uvular						r
Velar			k	g	ng	

Table A.4: French Vowels annotated in the corpus. '~' and '(X)' indicate nasal and approximant variants respectively.

Openness	Front		Central		Back	
	Unrounded	Rounded	Unrounded	Rounded	Unrounded	Rounded
Close	i (j)	y (h)				u (w)
Mid-Close	e	x	q			o
Mid-Open	e^/e~	x^/x~				o^/o~
Open	a/a~					

Appendix B

Tacotron2 Configuration

This appendix describes the procedure and hyperparameters used to train Tacotron2 in the presented contributions. Our implementation is publicly available online[1]. Modifications from the original Tacotron2 model [Shen et al., 2018] are detailed in Section 2.1. The baseline implementation for this work is one of the publicly available repository shared by NVIDIA[2].

This implementation allows by default the use of phonetic input sequences for training and inference, but the phonetic lexicon was adapted to our phonetic alphabet described in Appendix A.

B.1 Training Procedure

Unless stated otherwise, when included in an experiment, Tacotron2 models are trained from scratch for 100 epochs. Using mixed-inputs, the training corpus is presented twice by epoch: once with orthographic inputs and once with phonetic inputs. Orthographic and phonetic inputs are mixed randomly within each batch. The batch size is set to 64, and batches are randomly picked among utterances of approximate same length. This training takes roughly 100 hours of training on a single GPU Quadro RTX 8000[3]. Empirical observations indicate that the convergence of losses does not fully estimate the perceptual quality of syntheses produced by the model. 100 epochs was determined as a minimum to produce a synthesis quality comparable with audio samples shared by Shen et al. [2018][4].

The training phase starts with 10 epochs of coarse model initialization, during which the postnet is discarded and the learning rate is fixed at 10^{-3}. This phase enables the model to initiate the predictive process. Then, the postnet is reactivated and the learning rate decreases exponentially until reaching 10^{-5} at 90 epochs. During the last quarter of epochs, the fine-tuning of the gate loss is activated, with a multiplying factor $\lambda = 10$ (see Section 2.1.2 for further details).

[1] **Update needed**: https://github.com/MartinLenglet/Tacotron2
[2] https://github.com/NVIDIA/tacotron2
[3] 2 of these GPU are made available to the CRISSP team at GIPSA-lab.
[4] Listening page shared by Tacotron2 authors: https://google.github.io/tacotron/publications/tacotron2

B.2 Hyperparameters

Table B.1: Default hyperparameters of Tacotron2 in the presented studies. Red values indicate differences compared to the original implementation. Window Size and Hop Size values are given in ms considering a sampling rate of 22.05 kHz.

Model Part	Hyperparameters	Values (Original)
Encoder	# Input Symbols	131
	Symbol Embedding # Dim	512
	# Convolutional Layers	3
	- Direction	Phonemes Sequence
	- Kernel Size	5
	- # filters	512
	Bi-LSTM # Dim	512
Attention	LSTM # Dim	1024
	Location Sensitive # filters	32
	Features # Dim	128
	Dropout	0.1
Decoder	LSTM # Dim	1024
	Dropout	0.1
	# Frames by step	2 (1)
Prenet	Dim	128 (256)
	Dropout (active at inference)	0.5
Postnet	# Convolutional Layers	5
	- Direction	Temporal
	- Kernel Size	5
	- # filters	512
Mel-Spectrogram	# Coefficients	80
	Minimum frequency (Hz)	0
	Maximum frequency (Hz)	8 000
	Window Size (# samples / ms)	1024 / 46.44 ms
	Hop Size (# samples / ms)	256 / 11.61 ms

Appendix C

FastSpeech2 Implementation

This appendix describes the training procedure and hyperparameters used to train FastSpeech2 in the presented contributions. Modifications from the original FastSpeech2 model [Ren et al., 2021] are detailed in Section 2.3.1. Our implementation is based on one of the open source repository[1]. Similarly to Tacotron2, we use our phonetic alphabet described in Appendix A. The proposed implementation is publicly available online[2].

C.1 Training Procedure

Unless stated otherwise, when included in an experiment, FastSpeech2 is trained from scratch for 100 epochs. Using mixed-inputs, the training corpus is presented twice by epoch: once with orthographic inputs and once with phonetic inputs. Because of memory limitations, the batch size is set to 32 with FastSpeech2, compared to 64 with Tacotron2. Similarly to Tacotron2, the evaluation of losses on the validation set is not sufficient to assess the quality of the synthesis at inference. 100 epochs takes approximately 80 hours on one GPU Quadro RTX 8000 and produces synthesis quality comparable with audio samples shared by Ren et al. [2021][3].

When included in the corpus, the non-audio inputs (dictionary and homographs) are used at every stage of the training process. They are mixed with audio-inputs in each batch, with a ratio of 2/3 for audio inputs and 1/3 for non-audio inputs. While the training on non-audio inputs helps learning phonetic representations for rare words not seen in the audio corpus, this ratio minimizes the risks of degradation of the prosodic predictions and audio quality due to the absence of spectrogram-loss on the non-audio part of the corpus.

The learning rate was fixed to 10^{-3} during this first 32 epochs on the audio corpus[4]. After this initialization step, the learning rate exponentially decreased to reach 10^{-4} after 60 epochs.

[1] https://github.com/ming024/FastSpeech2

[2] This implementation was shared as part of the Blizzard Challenge 2023: https://github.com/MartinLenglet/Blizzard2023_TTS

[3] Listening page shared by FastSpeech2 authors: https://speechresearch.github.io/fastspeech2

[4] Following the 2/3 - 1/3 ratio, this training includes about 16 epochs on the non-audio corpus.

C.1.1 Multi-Speaker Training Procedure

We advocate for starting the training on the most seen speaker. We believe that this step helps the text encoder and decoder to focus on their primary goal which is the modulation of acoustic and prosodic local patterns according to the sequence to utter. The addition of the speaker embedding later in the process is seen as an offset manipulation of these mean features, which is supposedly easier to learn by the model.

Then, speakers are randomly mixed in each batch during training. The least seen speakers benefit from a final step of fine-tuning on an evenly distributed corpus for the last epochs. We empirically found that this final step helps modeling rarest speakers behaviors instead of copying the behavior of the most seen speaker.

C.2 Model Configuration

Hyperparameters used in all experiments presented in this manuscript (unless stated otherwise) are summed up in table C.1. The text encoder and the audio decoder are stacks of 4 and 6 Feed-Forward Transformer layers (FFT) respectively. The internal representations are set to 256 dimensions. Multi-head self-attention blocks in FFT layers have 2 heads.

Table C.1: Default hyperparameters of FastSpeech2 in the presented studies. Red values indicate differences compared to the original implementation. Window Size and Hop Size values are given in ms considering a sampling rate of 22.05 kHz.

Model Part	Hyperparameters	Values (Original)
Encoder	# Input Symbols	131
	Symbol Embedding # Dim	256
	# FFT Layers	4
	- # Heads	2
	- Conv1D Kernel Size	9
	- Conv1D # filters	1024
	Dropout	0.2 (0.1)
Variance Adaptor	Conv1D Kernel Size	3
	Conv1D # filters	256
	Dropout	0.5
Decoder	# FFT Layers	6 (4)
	- # Heads	2
	- Conv1D Kernel Size	9
	- Conv1D # filters	1024
	Dropout	0.2 (0.1)
Postnet	# Convolutional Layers	5 (0)
	- Direction	Temporal
	- Kernel Size	5
	- # filters	512
Mel-Spectrogram	# Coefficients	80
	Minimum frequency (Hz)	0
	Maximum frequency (Hz)	8 000
	Window Size (# samples / ms)	1024 / 46.44 ms
	Hop Size (# samples / ms)	256 / 11.61 ms

Appendix D

Vocoders Implementation

This appendix describes the implementation and training procedure of the vocoders used in the presented experiments. As stated in section. 1.1.6, the choice of neural vocoders depends on the trade-off between the optimal synthesis quality and the inference speed acceptable for the designed application. Although the experiments presented in this thesis evaluate pre-generated stimuli, we prioritized neural vocoders whose Real-Time-Factors (RTF) are superior to one (1s of waveform is generated in less than 1s), which better illustrate the capabilities of the tested algorithms in real-life situations. This excludes WaveNet [Van den Oord et al., 2016], which despite excellent audio quality, requires high computation power for a long period of time.

Three neural vocoders were consecutively used in the presented experiments. Motivations and implementations are described in the sections below. All audio waveforms and mel-spectrograms follow the same format, described in table D.1.

Table D.1: Default audio format used in the presented studies.

Representation	**Features**	**Values**
Mel-Spectrogram	# Coefficients	80
	Minimum frequency (Hz)	0
	Maximum frequency (Hz)	8 000
	Window Size (# samples / ms)	1 024 / 46.44 ms
	Hop Size (# samples / ms)	256 / 11.61 ms
Audio Waveform	Sampling Rate	22 050
	Recording Settings	Mono
	Quantization	32 bits

D.1 WaveRNN

WaveRNN [Kalchbrenner et al., 2018] is an autoregressive vocoder which produces the best audio quality among the systems evaluated by Govalkar et al. [2019].

The implementation of WaveRNN used is available online[1]. No modifications were made to

[1] https://github.com/fatchord/WaveRNN

this implementation. The model is trained from scratch for 1520 epochs on the single-speaker corpus used for the segmentation experiment reported in Table 3.1. During the first 1000 epochs, the learning rate is fixed to 10^{-4}. Then the learning rate is reduced to 10^{-5} for the remaining 520 epochs. The batch size is set to 32 for the whole training.

D.1.1 Limitations

WaveRNN's RTF of 1 makes it difficult to envision using it for real-time applications. We initially thought about adapting this autoregressive architecture in order to generate and play the audio waveforms in smaller chunks, which would have resulted in a more acceptable latency. This adaptation was too complicated to implement, and faster vocoders were favored in later works.

D.2 Waveglow

Waveglow [Prenger et al., 2019] is a parallel vocoder which combines a fast inference speed (RTF of 30 on GPU) with a good audio quality. This vocoder is better suited for applications that need real-time responses.

We use the implementation shared by NVIDIA[2]. We fine-tuned the pre-trained-model shared with the GitHub implementation[3]. The fine-tuning was performed on the NEB corpus (see table A.1), first for 50 epochs on the Ground-Truth spectrograms, and then for 50 additional epochs on spectrograms predicted by the Tacotron2 model trained for the first experiment described in section 3.2.

D.2.1 Limitations

Participants to the multiple experiments performed with Waveglow often criticized the background noise produced by this vocoder. In attempt to avoid the use of heuristics, no post-processing denoising methods were used to reduce this background noise. Additionally, Waveglow is inherently designed to take advantage of the parallel processing performed by GPUs. As a result, inference speed on CPU is prohibitive for any real-time applications.

D.3 HiFi-GAN

HiFi-GAN [Kong et al., 2020] is another example of parallel neural vocoder. HiFi-GAN combines the performances comparable to the best autoregressive vocoders with an unmatched

[2]https://github.com/NVIDIA/waveglow
[3]https://drive.google.com/file/d/1rpK8CzAAirq9sWZhe9nlfvxMF1dRgFbF/view

inference speed. Even with the smaller memory footprint's configuration, HiFi-GAN reports MOS of 4.05, compared to 4.02 for WaveNet and 3.81 for Waveglow. With this configuration, the inference speed is 1187 times faster than real-time on GPU, and 13 times faster than real-time on CPU. This makes this vocoder particularly fitted for low resources environments, like on-device applications. This vocoder was chosen for the Theradia application [Tarpin-Bernard et al., 2021].

We use the implementation shared by the authors[4]. We chose to train a model from scratch on our French Dataset reported in table A.1. The configuration V2 [Kong et al., 2020] is chosen for its trade-off between performances and audio quality. With this configuration, this vocoder can generate waveforms on a CPU 5 times faster than real-time.

The model is first trained for 70 epochs on single speaker setup (only NEB). Afterward, the vocoder is fine-tuned on mel-spectrograms predicted by a pre-trained multi-speaker FastSpeech2 model. This fine-tuning is made on a balanced dataset among the 5 speakers of the corpus. To avoid overfitting on the most seen speaker, the maximum number of utterances by speaker is set to 2500[5]. Utterances are randomly picked by speaker. The model is trained for 600 epochs on this balanced set. The batch size is set to 16 for the whole training. The learning rate is initially set to 2×10^{-4}, and is multiplied by 0.999 after each epoch.

D.3.1 Limitations

HiFi-GAN produces overall good quality syntheses, but some recurrent artifacts remain. Notably, a whistling noise often appears when producing high frequency sounds, like fricative consonants. A noise reduction post-processing is added to mitigate this effect [Sainburg, 2019]. This noise reduction improves quality but mentioned artifacts are still audible. Further improvements could be obtained by training the model for more iterations[6]. Fine-tuning the model on our newly recorded expressive dataset (additional data for speaker AD) should also improve the performances of the model on more variate prosodic patterns.

[4] https://github.com/jik876/hifi-gan

[5] At the time of this training, only 2500 utterances were available for AD, which motivated this limit.

[6] The presented setup corresponds to 570 000 training iterations, compared to 2 500 000 iterations for the pre-trained "UNIVERSAL" model shared by the authors.

D.4 Hyperparameters

Table D.2: Default hyperparameters of vocoders in the presented studies.

Vocoder	Hyperparameters	Values (Original)
WaveRNN	Upsample Factors	5, 5, 11
	RNN # Dim	512
	FC # Dim	512
	Compute # Dim	128
	# Residual Blocks	10
	- Output # Dim	128
Waveglow	# Flows	12
	# Groups	8
	# Early Every	4
	# Early Size	2
	# Coupling Layers	8
	- # Channels	256
	- Kernel Size	3
HiFi-GAN	Upsampling # Blocks	4
	- Rates	8, 8, 2, 2
	- Kernel Sizes	16, 16, 4, 4
	- Initial Channel	128
	Residual # Blocks	3
	- Kernel Sizes	3, 7, 11
	- Dilatation Sizes	1, 3, 5

Appendix E

Abacus Pause Control

Two abacuses are recorded through the calibration phase described in section 5.4.1.2, to predict the magnitude of the pause embedding bias according to the expected pauses proportion. This appendix presents the abacuses for $\boldsymbol{TC_P}$ and $\boldsymbol{FS}$.

Table E.1: Abacus between target proportion of pauses and corresponding pause embedding bias magnitude for $\boldsymbol{TC_P}$.

Target Proportion of Pauses (#silences/#word boundaries)	**Pause Embedding Bias Magnitude**
0	-1.20
1	-0.82
2	-0.68
3	-0.57
4	-0.43
5	-0.20
6	0.01
7	0.14
8	0.22
9	0.28
10	0.33
11	0.36
12	0.38
13	0.40
14	0.41
15	0.43
16-17	0.44
18	0.45
19-20	0.46
21	0.47
22-23	0.48
24-26	0.49
27-28	0.50
29-31	0.51
32-35	0.52
36-38	0.53
39-42	0.54
43-46	0.55
47-51	0.56
52-55	0.57
56-60	0.58
61-64	0.59
65-69	0.60
70-73	0.61
74-77	0.62
78-80	0.63
81-83	0.64
84-86	0.65
87-89	0.66
90-91	0.67
92-93	0.68
94	0.69
95	0.70
96	0.71
97	0.72
98	0.74
99	0.76
100	0.97

Table E.2: Abacus between target proportion of pauses and corresponding pause embedding bias magnitude for ***FS***.

Target Proportion of Pauses (#silences/#word boundaries)	**Pause Embedding Bias Magnitude**
0	-0.96
1	-0.73
2	-0.64
3	-0.55
4	-0.46
5	-0.35
6	-0.19
7	0.03
8	0.22
9	0.29
10	0.33
11	0.36
12	0.38
13	0.40
14	0.41
15	0.42
16	0.43
17-18	0.44
19	0.45
20-22	0.46
23-24	0.47
25-27	0.48
28-31	0.49
32-35	0.50
36-40	0.51
41-45	0.52
46-50	0.53
51-56	0.54
57-62	0.55
63-68	0.56
69-73	0.57
74-78	0.58
79-83	0.59
84-87	0.60
88-90	0.61
91-93	0.62
94-95	0.63
96	0.64
97	0.65
98	0.66
99	0.68
100	0.79

Attached Publications

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Impact of Segmentation and Annotation in French end-to-end Synthesis

Martin Lenglet, Olivier Perrotin, Gérard Bailly

Univ. Grenoble Alpes, CNRS, Grenoble-INP, GIPSA-lab, France
{martin.lenglet,olivier.perrotin,gerard.bailly}@grenoble-inp.fr

Abstract

Audio books are commonly used to train text-to-speech models (TTS), as they offer large phonetic content with rather expressive pronunciation, but number and sizes of publicly available audio books corpora differ between languages. Moreover, the quality and accuracy of the available utterance segmentations are debatable. Yet, the impact of segmentation on the output synthesis is not well established. Additionally, utterances are generally used individually, without taking advantage of text level structuring information, even though they influence speaker reading. In this paper, we conduct a multidimensional evaluation of Tacotron2 trained on different segmentations and text level annotations of the same French corpus. We show that both spectrum accuracy and expressiveness depend on the segmentation used. In particular, a shorter segmentation, in addition with the annotation of paragraphs, benefits to spectrum reconstruction at the detriment of phrasing. Multidimensional analysis of mean opinion scores obtained with a MUSHRA-experiment revealed that phrasing was relatively more important than spectrum accuracy in perceptual judgement. This work serves as evidence that particular attention must be given to models evaluation, as well as how to use the training corpus to maximize synthesis characteristics of interest.
Index Terms: Speech Synthesis, French TTS, mixed-inputs TTS, French dataset

1. Introduction

In recent years, deep learning met huge success in language-related applications. In particular, state-of-the-art text-to-speech (TTS) models [1, 2, 3] coupled with neural vocoders [4, 5] achieve synthesis quality close to natural speech. As always with deep learning, the quality of the output heavily depends on the dataset used for training. The common approach of neural TTS, seen in events like Blizzard Challenge [6], is to compare multiple models on the same corpus to evaluate the resulting synthesis quality. This process minimizes the importance of input data structuring, which ultimately shapes the output of any deep learning model. One complementary work is to evaluate multiple segmentations of data structuring on the same TTS model. This paper adopts this approach.

Publicly available corpora designed to train TTS [7, 8, 9] are generally composed of audio book extracts read by one or more speakers, segmented in thousands of utterances. Utterances' lengths vary between 1 to 20 seconds, with boundaries often matching sentences, but not always. Even if these databases have been used to train state-of-the-art speech models [3, 10], long utterances may not be the best candidates to train TTS: (i) Large batch size with long utterances rely on high computation memory. (ii) Learning long-term dependencies is a challenging task for sequential models [11]. (iii) Style control, which is an increasing demand of the field, massively uses utterance level style embeddings [12, 13], which means that the shorter the utterances, the finer it is possible to tune speech style at inference time. These reasons made us consider a shorter segmentation may be better suited to train TTS efficiently.

Proposing a new segmentation gives us the opportunity to integrate specific annotations in the input data to give models relevant context information regarding the corresponding speech to produce: (i) End of paragraph are generally associated with specific phrasing modifications from the speaker, and are then worth noticing during training. (ii) In French, silent letters and optional liaisons are common, which are additional difficulties to train a TTS model on orthographic inputs alone. The addition of phonetic annotations contributes to alleviate this issue, and has shown to benefit to both transcriptions [14].

This paper presents a multidimensional comparison between the proposed segmentation and annotation of the LibriVox French corpus [15] and the original segmentation from M-AILABS [7], used to train the same Tacotron2 [1]. We evaluate the phrasing and spectral accuracy of each model. These objective measurements are paired with mean opinion scores evaluated through a MUSHRA-like experiment [16].

2. Related Work

To our knowledge, there is no publicly available French Tacotron2. Recent studies published on French synthesis focus on concatenation based TTS [17] or use Deep Convolutional TTS (DCTTS) [18]. DCTTS is a fully convolutional neural TTS, whose initial purpose was to alleviate the need for high computational power, while enabling quick training on smaller database. Although synthesis reaches acceptable standards, the overall quality does not match more recent models [1, 2, 3].

The later TTS explore the well established encoder-decoder architecture: the encoder converts the input sequence into a hidden representation that the decoder uses to generate mel-spectrogram frames. As an interface between the two, Tacotron2 [1] employs a location-sensitive attention [19] module which computes a fixed length vector for each decoder step. The encoder adopts an approach that is similar to the classical language model processing pipeline: the input sequence is passed through three convolutional layers that compute local pattern, followed by bidirectional LSTM. Alternatively, Transformer TTS [2] and Fastspeech [3] introduce self-attention and multi-head attention layers as a replacement for recurrent units. These three models produce synthetic speech of similar quality [3]. We chose Tacotron2 for its relative ease to implement and straight training process. Additionally, Tacotron2 shows promising results for expressive control [12, 13], which is also one of our short term goal.

Although mean opinion scores are generally used to assess the global quality of TTS, this evaluation takes multiple aspects of speech into account: phonetic correctness and intelligibility,

10.21437/SSW.2021-3

spectral smoothness, expressiveness, etc. These clues may not vary conjointly, which means that the use of a single metric may not be sufficient. [20, 21] employ multidimensional scaling (MDS) [22] to extend the quality analysis of TTS models. This paper prolongs this perspective.

3. Proposed Method

This section presents the original baseline and the new segmentation proposed from the French LibriVox dataset, and the modifications added to the Tacotron2 implementation shared by NVIDIA[1]. Our implementation[2] and database[3] are available online.

3.1. Segmentation and Annotation

3.1.1. Original Database

We used the M-AILABS French dataset [7] as a starting point. This corpus includes more than 190h of recorded speech, segmented in utterances from 1s to 20s, given with corresponding orthographic transcripts. Recordings come from the free public domain audio books LibriVox database [15]. We selected a subset of the recordings made by Nadine Eckert-Boulet (NEB), for a total duration of 34h. Each book duration and corresponding number of utterances are given in Table 1. Audio files are originally sampled at 16000Hz, but we re-sampled them at 22050Hz.

Table 1: *Books duration (and number of utterances) for original and new segmentation of the M-AILABS French corpus.*

Book	**Original**	**New segmentation**
Les Mystères de Paris	22:31:27 (12285)	21:37:21 (25458)
Mme Bovary	11:39:50 (5775)	11:08:55 (12781)
Total	34:11:17 (18060)	32:46:16 (38239)

The orthographic transcript is given by the Gutenberg Project[4]. It is worth mentioning that NEB does not always strictly follow the original text. Some miss-spelling remain (for example: "precepteur" is said instead of "percepteur"), as well as some omissions. These miss-alignments correspond to 0.1% of the original corpus. We did not correct any of those transcripts for the baseline. Though, we spelled out all texts, including frequently used abbreviations in French ("M.": "Monsieur", "Mlle": "Mademoiselle", "n°": "numéro" and "etc": "et cetera"), and numbers ("1838": "dix-huit cent trente-huit"). Two punctuation marks were also replaced to stand as a single unique character: "..." was replaced by "$\sim$", "–" by "¬".

Each clip was originally bounded with 500ms of silence (zeros in the waveform) at the beginning and the end. These silences do not correspond to the recordings, but have been artificially added to each audio clip after segmentation. To limit the duration of initial and final silences in the synthesis, we truncated these silences at 130ms. This duration matches the initial and final silence lengths found in other speech databases such as LJspeech [8].

3.1.2. Re-segmentation

To reduce the average duration of utterances, we first restore the initial audio books chapters structure by aligning the original text from the Gutenberg Project with the recordings from LibriVox. As for the original segmentation, all texts are spelled out, but previously mentioned miss-spelling and omissions are now manually corrected. In addition, end of paragraphs are annotated with the punctuation mark "§", which is introduced after the last punctuation mark preceding each carriage return. Ends of paragraphs are accompanied by phrasing patterns of NEB, that are worth highlighting in the training corpus. For instance, Table 2 shows F0 and elongation of the final syllable before ends of paragraph vs. paragraph-internal periods, as well as their values for the following syllable. The last syllable is generally longer before the end of paragraph, and the F0 gap across the boundary is increased (6.45 vs. 5.11 semitones respectively).

Table 2: *Comparison of F0 and elongation of syllable [23] around ends of paragraph (.§) and intermediate periods (.).*

		Syllable	
		Previous	**Following**
Elongation (%)	**.**	+184	+21
	.§	+218	+24
F0 (semitone)	**.**	1.96	7.01
	.§	0.96	7.41

Chapters are then segmented based on silences of at least 400ms. This duration usually corresponds to pauses made between speaking turns in conversations [24]. 94.56% of silences coincide with punctuation marks. For the others, a comma is added at the end of the utterance. 130ms of ambient silence from the recording are kept at the beginning and the end of each utterance. Timestamps were hand-checked for each utterance to ensure optimal segmentation. Table 1 shows duration and number of utterances of the obtained segmentation. Note that the proposed segmentation is 01:25:01 shorter than the original, due to the reduction of intra-utterance silences, but that reduction does not impact either the text read nor the speaking rate.

Fig.1 gives the distribution of utterances length of the original and the proposed segmentation. Median utterance length (resp. first and third quartiles) are reduced from $6.44s$ ($3.88s$ and $9.26s$) to $2.77s$ ($1.89s$ and $3.95s$). 82.5% of utterances of the new segmentation last between $1s$ and $5s$, and 0.25% of utterances last more than $10s$. 1336 utterances are unchanged, which corresponds to 7.4% and 3.5% of the original and new segmentation respectively.

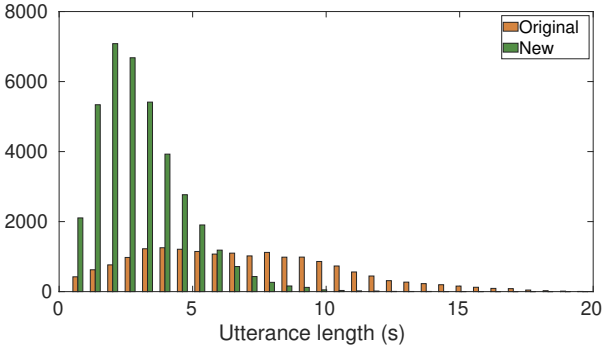


Figure 1: *Distribution of utterances length of original and new segmentation.*

[1] https://github.com/NVIDIA/tacotron2
[2] https://github.com/MartinLenglet/Tacotron2
[3] https://zenodo.org/record/4580406#.YI_qIyaxXmE
[4] https://www.gutenberg.org/

3.1.3. Phonetic Annotation

Training of both orthographic and phonetic transcripts, called representation mixing, enables to use both input types in the same utterance at inference time, and thus remove some ambiguities on particular issues, without the need for the whole phonetic transcript of the speech to synthesize. For instance, NEB performs numerous optional liaisons (22999 liaisons in the corpus of which 9597 [z], 9029 [t] and 3412 [n]), in particular bridging 844 infinitives and prepositions with [ʁ/]. Yet, these liaisons are not systematic, and adding the possibility to choose if the liaison is being made at inference time (as part of a style component) would be interesting. To study the impact of phonetic annotation, hand-crafted phonetic alignment is performed on the whole new segmentation.

3.2. Modifications of Tacotron2

3.2.1. Representation mixing

We introduce the mixed embedding matrices described in [14] in our model to give the possibility to train with both types of inputs. Contrary to [14], when the training includes phonetic inputs, input types are not mixed within the same utterance. The number of utterances is simply doubled, with the same audio file corresponding to both the orthographic and the phonetic input.

3.2.2. Gate loss correction

Synthesizing short utterances, typically one or two words, has been shown to be a challenging task for TTS models [25]. Recurrent artifacts are repetition of the last syllable, or unintelligible words. With our proposed segmentation, 5% of utterances last less than 1s, which might cause some issues during inference. To avoid this, we fine tune the training of each model with 2 modifications: (i) 9 frames of recorded ambient silence are added at the end of each utterance, in which the end-of-sequence probability is set to 1. This silence originates from the pause following each utterance. (ii) a multiplying factor is added to the gate loss error before back-propagation. We empirically found that these modifications correct previously mentioned artifacts, and improve the overall synthesis quality. The benefits of these modifications are evaluated in section 4.

4. Experiments and Results

4.1. Experimental Setup

The 6 models trained for this experiment are presented below:

- O **and** O_g are trained on the original segmentation from M-AILABS for 200 epochs.
- N **and** N_g are trained on the new segmentation proposed in section 3.1.2, with only orthographic inputs for 200 epochs.
- P **and** P_g are trained on the new segmentation proposed in section 3.1.2, with both orthographic and phonetic inputs for 100 epochs, since each epoch corresponds to twice the number of utterances of the orthographic models.

Models annotated $_g$ are fine-tuned with the gate loss correction. The multiplying factor is set to 10 for these models. This correction is introduced for the last quarter of the training epochs. Before that separation, only one model is trained using warm-start from the English model trained on LJSpeech shared by NVIDIA. The postnet is bypassed during the first 10 epochs, and the learning rate is fixed at 10^{-3}. This phase enables the model to initiate a coarse transition from English to French. Then the postnet is reactivated and the learning rate decreases exponentially until reaching 10^{-5} at 90 epochs. The batch size is limited to 32, due to memory limitations with long utterances of the original segmentation, and thus is set to 32 for all models. Batches are randomly picked among utterances of approximate same length.

We pick 5% of the original corpus as test set. To ensure a fair comparison between models, these 903 utterances are randomly selected among the 1336 common utterances between the original and the new segmentation. Thus, the amount of speech seen by each model during training is rigorously the same. Only the orthographic transcript of the test set is used in this section, even for models P and P_g. Note that this test set does not favor the new segmentation: phonetic inputs and paragraphs markers are not used.

The vocoder used is WaveRNN [5]. WaveRNN is faster and demands less resources than the original WaveNet [4] used by [1], and still provides a good voice quality [26]. We trained WaveRNN from scratch for 1000 epochs on the new segmentation from Table 1 with a learning rate of 10^{-4}. Then we fine-tuned the model with 520 more epochs at a learning rate of 10^{-5}.

4.2. Objective measurements

4.2.1. Accuracy

We evaluate the spectral accuracy of each model through the proximity of the generated spectra with the *vocoded* ground truth (GT). Since syntheses differ in length, mel-spectrograms are first aligned by dynamic time warping (DTW) [27]. Mean squared error (MSE) on aligned spectrograms are then computed and averaged on the test set; results are shown in Fig. 2.

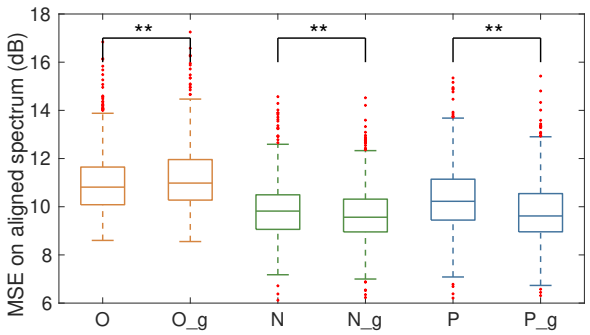


Figure 2: *Mean squared error between models and ground truth, calculated on mel-spectrograms aligned by dynamic time warping. ** indicates a significant effect of the gate loss correction according to Tukey-Kramer test ($p < 0.05$).*

The model has a statistical effect on the computed distances according to a one-way ANOVA ($F = 246.5$, $p < 0.001$). Tukey-Kramer multiple comparisons show that all pairs are statistically different, except P_g/N and P_g/N_g. The gate loss correction has a significant impact on all models. The new segmentation decreases the spectral distortion, with a beneficial contribution of the gate loss correction in this case. On the other hand, this correction decreases the spectral accuracy of the model trained on the original segmentation.

4.2.2. Phrasing

Pauses position and duration contribute to the expressiveness of speech [28]. We computed mean speech and silence duration across the whole synthesised test set for each model and for

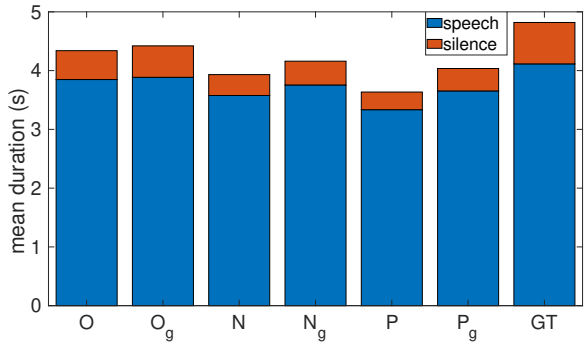


Figure 3: *Mean utterance duration on the whole test set for each model.*

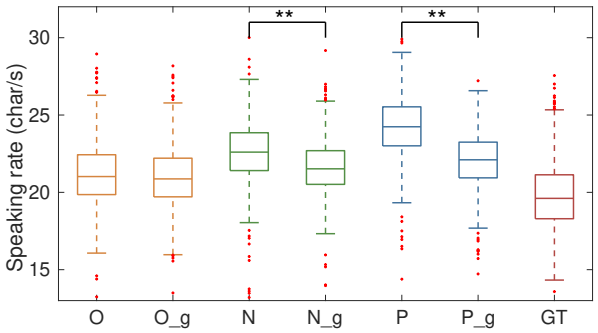


Figure 4: *Speaking rate of each model, calculated on each utterance of the test set. Speaking rate is estimated in characters per second, pause durations are not taken into account. ** indicates a significant effect of the gate loss correction ($p < 0.05$).*

GT. By extension, this calculation also enables us to estimate the speaking rate of each model on the test set. Mean utterance duration and speaking rate are shown in Fig. 3 and Fig. 4 respectively.

Models trained on the new segmentation do not exhibit the same temporal behavior than models trained on the original segmentation. Utterances mean duration is smaller with the new segmentation ($3.93s$ and $3.64s$ compared to $4.44s$ for N, P and O respectively). Silences duration are also proportionally smaller: 9.2%, 8.2% and 11.3% for N, P and O respectively. As a result, the speaking rate increases with the new segmentation. Note that the speaking rate of all models is significantly higher than GT. The gate loss correction tends to reduce the differences observed compared to GT. Not only silences duration are increased, but also speech duration, resulting in a lower speaking rate. This decrease is statistically significant for the new segmentation, but not for the original. All other pairs are significantly different according to Tukey-Kramer multiple comparisons.

Longer pauses observed with O and O_g may result from the intra-utterance pauses frequency and duration in the original segmentation provided by M-AILABS. In that case, models are trained on audio clips that sometimes contain pauses longer than 1s, and thus reproduce that behavior during inference. On the contrary, the re-segmentation processing avoids intra-silences longer than 400ms, resulting in a more straight-forward synthesis.

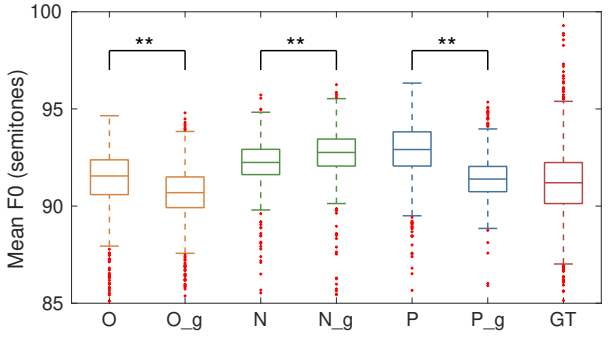


Figure 5: *Mean fundamental frequency calculated on voiced sections of each utterance of the test set. ** indicates a significant effect of the gate loss correction ($p < 0.05$).*

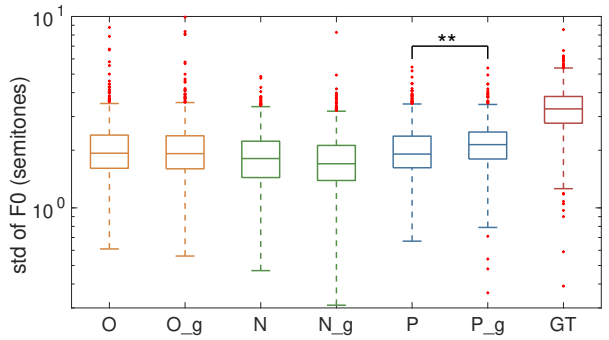


Figure 6: *Standard deviation of fundamental frequency calculated on each utterance of the test set. ** indicates a significant effect of the gate loss correction ($p < 0.05$).*

4.2.3. Pitch

As additional prosody measurements, we evaluate the pitch of each model using the Praat software [29]. The mean fundamental frequency (F0) and standard deviation of F0 is measured on voiced sections for every utterance of the test set. Results are given in Fig. 5 and Fig. 6 respectively.

One-way ANOVA shows a statistical effect of the model on both mean F0 and standard deviation of F0. Regarding mean F0, Tukey-Kramer multiple comparisons show that all pairs differ significantly, except O/P_g, O/GT and N_g/P. As to standard deviation of F0, only phonetic models P and P_g exhibit a significant effect of the gate loss correction, while both P and P_g are not statistically different from O and O_g. N and N_g have significantly lower standard deviation than all other models.

The new segmentation increases mean F0, but this effect is partially compensated when training the model on mixed inputs with gate loss correction. Similarly, the gate loss correction induces a lower mean F0 when training on the original segmentation. None of the presented models show standard deviation of F0 similar to GT, which might lead to less expressive synthetic voices.

4.3. Subjective evaluation

In accordance with objective measurements presented in section 4.2, 3 models were selected to evaluate the mean opinion scores through a MUSHRA-like experiment [16]. We keep only models that have been fine-tuned with gate loss correction, as they generally exhibit the closest proximity with GT behavior.

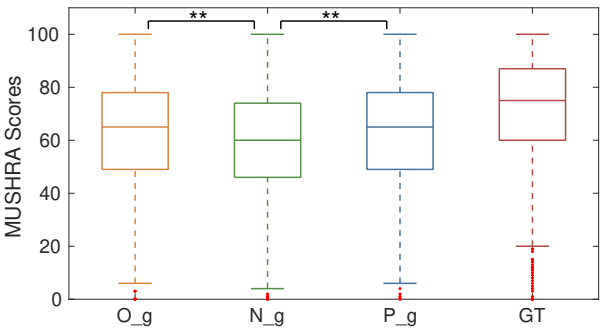
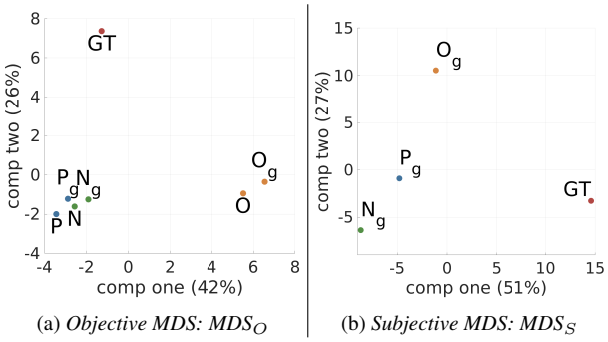


Figure 7: *MUSHRA results. ** indicates a significant difference between models ($p < 0.05$).*



(a) *Objective MDS: MDS_O* (b) *Subjective MDS: MDS_S*

Figure 8: *Multidimensional scaling of distances between pairs conditions. Left and right graphs show objective and subjective distances respectively. Proportions of variance explained are given for each component.*

GT is added as high anchor for the MUSHRA. This perceptive test was performed online using the webMUSHRA framework [30]. Utterances containing less than 7 words and more than 23 words were excluded from this test to keep only the central 90% of the test set length distribution. 60 utterances were randomly selected in the remaining test set, with equivalent representation of utterance lengths in the selection. 13 of the selected utterances contained one phonetic mistake (5 in O_g, 3 in P_g, and 5 in all models), and were replaced before the experiment. Participants were separated in 2 groups, each group listened to 30 out of the 60 selected utterances. For each utterance, participants were given the original text input, and were asked to evaluate the 4 given conditions (3 models + GT) according to the voice quality. No explicit reference was given during the listening. The experiment began with 5 minutes of training during which participants listened to a variety of synthesis that they were about to hear during the experiment and learned how to use the webMUSHRA interface. Audio examples are available online [5]. 44 participants recruited on Prolific [31] and aged 18-65 took part in the experiment. Participants were French native speakers, and had little or no previous experience with listening tests. Results of the MUSHRA are given in Fig.7

We compared the median score of each model using a Wilcoxon rank sum test. Differences are significant if $p < 0.05$. GT exhibits a significantly higher score than the 3 evaluated models. N_g scores significantly lower than all other models. No statistical differences are shown between O_g and P_g.

4.4. Multidimensional analysis

Despite the differences on specific expressiveness clues measured in section 4.2, subjective evaluation performed in section 4.3 does not exhibit a clear perceptive preference for one of the models O_g or P_g. To explore implicit dimensions of the evaluation of the models, we use a multidimensional analysis of the distances computed between each model and GT. These distances are evaluated on both objective and subjective measurements:

- **Subjective distances:** absolute score differences between all possible condition pairs evaluated in the MUSHRA, averaged across all participants and all utterances.
- **Objective distances:** MSE between all possible conditions pairs computed on mel-spectrograms aligned by DTW [27]. Objective distances are averaged across all 903 utterances of the test corpus.

[5] http://www.gipsa-lab.fr/~martin.lenglet/segmentation_impact/index.html

Table 3: *Correlation coefficients between objective measurements and components of MDS. * and ** indicate $p < 0.1$ and $p < 0.05$ respectively. ASE: aligned spectrum error, SR: Speaking rate, PD: pauses duration.*

MDS	**Dim**	**objective measurements**				
		ASE	SR	PD	mean F0	std F0
Obj	1	**0.90****	-0.47	0.44	**-0.71***	-0.06
	2	0.63	**-0.74***	**0.89****	-0.43	**0.97****
Subj	1	0.89	**-0.93***	**0.96****	-0.50	**0.98****
	2	0.97	-0.02	0.13	-0.83	-0.17

Then, we projected the two obtained distances matrices in two independent 2-dimensions space using classical Multidimensional scaling (MDS) [22]. To give a better idea of the impact of the gate loss correction, both corrected and non-corrected models were included in the objective MDS. Subjective and objective MDS (named MDS_S and MDS_O respectively in the following) are given in Fig.8.

Correlations between objective measurements computed in section 4.2 and the components of both MDS are estimated. Correlations coefficients are given in Table 3. Note that GT is not considered for correlation with aligned spectrum error (ASE). Correlation coefficients indicate that prosodic clues like pauses duration and standard deviation of F0 are closely related to the second component of MDS_O, but to the first component of MDS_S. On the other hand, spectral accuracy measurements ASE and mean F0 are correlated to the first component of MDS_O, and similarly for the second component of MDS_S, even if this tendency is not significant. Two main dimensions emerge in both evaluations: spectrum accuracy and expressiveness. The axis inversion (and associated portion of variance explained) tends to show these dimensions are not given the same importance in the perceptive judgement than in the objective measurement. As a result, the proximity of spectrum quality observed between GT and models trained on new segmentation on the first component of Fig.8a is downgraded to the second component of Fig.8b. Respectively, expressiveness is given more importance in the perceptive test than it is in the objective measurements, resulting in O_g being closer to GT in the first component of Fig.8b. Fig.8a emphasizes the benefits of the proposed gate loss correction, as all models annotated $_g$ are closer to GT on the expressiveness dimension.

5. Conclusions and Discussion

We have proposed a shorter segmentation of the French M-AILABS corpus and compared the training of Tacotron2 on both original and new datasets. Through multi dimensional evaluation, we have shown that the way speech data are segmented impacts both quality and expressiveness factors in opposite directions. Future works should elaborate on how to combine the advantages of both segmentation with curriculum training. An important contribution of this work is the addition of the gate loss correction as a fine tuning of the model, which contributes to improve prosodic aspects of the synthesized speech. The use of multidimensional analysis of mean opinions scores introduces relevant nuances to the MUSHRA results. The structuring of the subjective notation latent space, as well as the prediction of positions in this space thanks to objective measurements should be the focus of future works.

6. Acknowledgments

This research has received funding from the BPI project THERADIA and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). This work was granted access to HPC/IDRIS under the allocation 2021-AD011011542R1 made by GENCI.

7. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[3] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[4] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[6] X. Zhou, Z.-H. Ling, and S. King, "The blizzard challenge 2020," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 1–18.

[7] I. Solak, "The M-AILABS speech dataset," https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/, 2019.

[8] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[9] P.-E. Honnet, A. Lazaridis, P. N. Garner, and J. Yamagishi, "The siwis french speech synthesis database. design and recording of a high quality french database for speech synthesis," Idiap, Tech. Rep., 2017.

[10] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[11] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A field guide to dynamical recurrent neural networks*, Y. Hochreiter, Sepp Bengio, P. Frasconi, J. Kolen, and S. Kremer, Eds. IEEE Press, 2001.

[12] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[13] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP*. IEEE, 2019, pp. 6945–6949.

[14] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.

[15] J. Kearns, "Librivox: Free public domain audiobooks," *Reference Reviews*, 2014.

[16] I. BS, "1534-1, method for the subjective assessment of intermediate quality level of coding systems," *International Telecommunications Union, Geneva, Switzerland*, vol. 14, 2003.

[17] M. Shamsi, J. Chevelu, N. Barbot, and D. Lolive, "Corpus design for expressive speech: impact of the utterance length," in *Speech Prosody*. ISCA, 2020, pp. 955–959.

[18] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *ICASSP*. IEEE, 2018, pp. 4784–4788.

[19] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[20] C. Mayo, R. A. Clark, and S. King, "Multidimensional scaling of listener responses to synthetic speech," in *Interspeech*. ISCA, 2005, pp. 1725–1728.

[21] F. Hinterleitner, S. Möller, C. Norrenbrock, and U. Heute, "Perceptual quality dimensions of text-to-speech systems," in *Interspeech*, 2011.

[22] J. B. Kruskal, *Multidimensional scaling*. Sage, 1978, no. 11.

[23] P. Barbosa and G. Bailly, "Characterisation of rhythmic patterns for text-to-speech synthesis," *Speech Communication*, vol. 15, no. 1, pp. 127–137, 1994.

[24] G. Bailly and C. Gouvernayre, "Pauses and respiratory markers of the structure of book reading," in *Interspeech*, 2012, pp. 2218–2221.

[25] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[26] P. Govalkar, J. Fischer, F. Zalkow, and C. Dittmar, "A comparison of recent neural vocoders for speech signal reconstruction," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 7–12.

[27] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.

[28] E. Godde, G. Bailly, D. Escudero, M.-L. Bosse, and E. Gillet-Perret, "Evaluation of reading performance of primary school children: Objective measurements vs. subjective ratings," in *International workshop on child computer interaction (WOCCI)*, 2017.

[29] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, pp. 341–345, 01 2001.

[30] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webmushra—a comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, no. 1, 2018.

[31] S. Palan and C. Schitter, "Prolific. ac—a subject pool for online experiments," *Journal of Behavioral and Experimental Finance*, vol. 17, pp. 22–27, 2018.

XXXIVe Journées d'Études sur la Parole – JEP 2022
13-17 June 2022, Île de Noirmoutier, France

Modélisation de la Parole avec Tacotron2 : Analyse acoustique et phonétique des plongements de caractère

Martin Lenglet Olivier Perrotin Gérard Bailly
Univ. Grenoble Alpes, CNRS, Grenoble-INP, GIPSA-lab, Grenoble, France
martin.lenglet,olivier.perrotin,gerard.bailly@grenoble-inp.fr

RÉSUMÉ

Les réseaux de neurones bouleversent depuis plusieurs années les applications de traitement automatique de la parole. Cependant, le bon de performances rendu possible par ces technologies se fait généralement au détriment de la compréhensibilité et de l'interprétabilité de ces nouveaux modèles. Pourtant, l'apprentissage statistique, au coeur de ces nouveaux usages, constitue une source potentielle d'informations importante sur le langage, à condition de réussir à identifier et localiser ces paramètres dans des réseaux de plusieurs millions de neurones. Ce papier propose une étude des plongements internes d'un modèle de synthèse vocale de type Tacotron2 entraîné sur le Français. Cette analyse suggère que le réseau apprend à représenter sa séquence d'entrée en une suite de cibles acoustiques et phonétiques, dépendantes de leur contexte. La mise en évidence de l'encodage de ces paramètres permet d'imaginer leur contrôle de manière plus naturelle.

ABSTRACT

Language Modeling with Tacotron2 : a Phonetic and Acoustic Analysis of Text Embeddings

In recent years, deep learning breakthroughs met a huge success in automatic speech processing. However, the leap forward in performances is accompanied by a lack of interpretability and understandability of these new models. Nevertheless, statistical learning constitutes a valuable source of information about language if analyzed with the right tools and methodology. This paper presents a study of text embeddings computed by a state of the art synthesis model Tacotron2 trained on French data. Our analysis shows that this network is able to compute an in-context sequence of acoustic and phonetic targets from the given input sequence. Identification and localization of these acoustic parameters would enable a more natural control over the synthesis.

MOTS-CLÉS : Synthèse de parole, réseau de neurones, plongement, réseau avec attention, transcription phonétique.

KEYWORDS: Speech synthesis, neural network, embeddings, attention network, phonetic transcription.

1 Introduction

Ces dernières années, les modèles d'apprentissage profond ont révolutionné l'approche du traitement automatique de la parole. En particulier, la synthèse de parole, portée par des modèles tels que Tacotron2 (Shen *et al.*, 2018) ou FastSpeech2 (Ren *et al.*, 2020), atteint une qualité proche de la voix naturelle. Tous ces modèles ont une architecture commune : un encodeur convertit la séquence d'entrée (orthographique et/ou phonétique) en une représentation abstraite, qu'un décodeur lit pour générer la séquence de trames de mel-spectrogramme correspondante.

10.21437/JEP.2022-83

Différentes stratégies de représentation des éléments de la séquence d'entrée sont proposées dans la littérature. Tacotron2 (Shen *et al.*, 2018) associe par exemple 3 couches de convolutions d'empan limité (2 caractères de chaque côté) à une unité Long-Short-Term Memory (LSTM) bidirectionnelle. Dans Fastspeech2 (Ren *et al.*, 2020), la couche récurrente LSTM est remplacée par plusieurs couches de self-attention. Dans les deux cas, la volonté est de mettre en contexte ces plongements afin de préparer la génération du signal audio associé.

Cependant, l'apprentissage automatique des milliers de paramètres qui constituent ces couches de neurones empilées, sans autres contraintes que la structure imposée au réseau, complique la lecture des paramètres encodés dans ces représentations latentes. Dans le même temps, l'apprentissage statistique est la force de ces modèles, qui parviennent à extraire les invariants dans les données d'apprentissage, ainsi que les covariations entre ces paramètres. Identifier et localiser ces paramètres permettrait tout d'abord de développer une méthodologie d'analyse linguistique de la parole d'un ou plusieurs locuteurs basée sur un apprentissage statistique sur de larges corpora. L'analyse des représentations latentes apprises par le modèle pourrait également permettre de gagner en contrôle sur la synthèse, en biaisant le modèle tout en respectant les covariations apprises entre les paramètres acoustiques d'intérêt. Cet article présente notre analyse linguistique de l'espace latent en sortie de l'encodeur d'un modèle Tacotron2. La section 2 présente le contexte de l'étude proposée. L'ensemble des méthodes nécessaires à l'analyse de l'espace latent sont décrites en section 3. L'analyse des représentations phonétiques des entrées orthographiques apprises par le modèle est détaillée en section 4.2. La section 4.3 développe l'analyse acoustique des paramètres encodés dans les représentations de la séquence d'entrée donnée au modèle.

2 Visualisation des plongements dans la littérature

La question de l'information encodée dans les plongements appris par un réseau d'apprentissage profond et la visualisation de ces paramètres est au coeur de la volonté naissante d'explicabilité de ces nouvelles méthodes de traitement de l'information (Burkart & Huber, 2021). L'exploration manuelle de l'espace latent en sortie d'un encodeur de type Tacotron (Wang *et al.*, 2017) entraîné sur des entrée orthographiques a montré une tendance à la représentation phonétique de ces entrées (Perquin *et al.*, 2020). Les auteurs montrent par visualisation t-SNE que les plongements se rassemblent par groupes de phonèmes, chaque groupe rassemblant des plongements d'un ou plusieurs graphèmes, qui dans leur contexte correspondent à ce phonème commun.

A l'échelle de la phrase complète, l'analyse des plongements de style ou de locuteur, massivement utilisés en synthèse expressive multi-locuteurs, s'inscrit dans cette même démarche. La visualisation par UMAP des plongements de style appris par encodeur de référence (Skerry-Ryan *et al.*, 2018) combiné à un modèle DCTTS (Tachibana *et al.*, 2018) a mis en évidence les corrélations entre les dimensions de l'espace réduit et certains paramètres acoustiques d'intérêt pour le contrôle expressif (Tits *et al.*, 2019). De même, l'observation des plongements de locuteur (Hsu *et al.*, 2019) révèle une organisation de l'espace latent par proximité vocale entre les locuteurs, séparant notamment les voix masculines des voix féminines.

Bien que ces travaux mettent en lumière l'information acoustique et phonétique potentiellement encodée dans l'espace latent des plongements, les méthodes non-linéaires de transformation de l'espace utilisées ne permettent pas de localiser les paramètres acoustiques dans l'espace latent initialement construit par le modèle. Nous souhaitons ainsi prolonger cette analyse pour transformer ces espaces latents en espaces de contrôle opérationnels pour la synthèse.

3 Méthodes proposées

Cette section décrit les méthodes d'analyse de l'espace latent en sortie de l'encodeur Tacotron2 utilisées dans ce papier. Dans l'objectif de vérifier que les entrées orthographiques sont interprétées de façon phonétique par l'encodeur, l'ensemble des méthodes décrites dans cette section sont appliquées sur des séquences d'entrées orthographiques.

3.1 Lecture de la carte d'attention

Dans Tacotron2 (Shen *et al.*, 2018), un réseau avec attention (Bahdanau *et al.*, 2014) réalise l'interface entre l'encodeur et le décodeur. Pour chaque trame de mel-spectrogramme générée de façon auto-regressive par le décodeur, cette couche d'attention calcule le poids assigné à chaque élément de la séquence d'entrée. Ces poids, appelés poids d'attention, indiquent l'importance relative des éléments de la séquence d'entrée pour la trame à générer. Nous faisons donc l'hypothèse que la lecture de ces poids permet de réaliser la segmentation automatique du spectrogramme en sortie en fonction des graphèmes donnés en consigne. Notre méthode de lecture de la carte d'attention s'applique à tous les caractères de la séquence d'entrée selon la procédure ci-dessous :

1. Vérification que le poids d'activation maximum de ce caractère sur la séquence dépasse un seuil fixé à 0.35. Ce seuil permet d'exclure les caractères qui ne sont pas suffisamment représentés dans le vecteur de contexte calculé par la couche d'attention. Ces caractères sont qualifiés de "muets" dans la suite de cet article.
2. Sur la trame pendant laquelle ce maximum est atteint, on vérifie que le poids d'attention maximum est sur le caractère en question. Sinon, on considère également ce caractère comme muet.
3. Si le poids est maximum sur le caractère, on considère que l'attention de la trame se porte sur ce caractère, de même pour toutes les trames adjacentes pour lesquelles le poids d'attention est maximum sur ce caractère.

Cette méthode permet d'estimer la durée d'attention de chaque caractère, ainsi que les trames pendant lesquelles ce caractère a été généré. Les performances de cette méthode sont évaluées en section 4.3.1.

3.2 Identification des schémas d'activation des graphèmes

La lecture des cartes d'attention de Tacotron2 a montré que tous les graphèmes n'étaient pas ciblés par l'attention. En effet, la correspondance graphème-phonème n'est généralement pas de 1 pour 1. Dans les langues à orthographe opaque comme le français, la prononciation correcte d'une lettre nécessite la connaissance d'un large empan de lettres (Bosse & Valdois, 2009). Un phonème est donc souvent associé à plusieurs graphèmes ; ce phonème est alors appelé phonème complexe. Dans ce cas, on note que l'attention du modèle se porte sur un seul graphème, selon un schéma qui dépend du contexte.

Les schémas les plus courants sont résumés en figure 1, et les règles de correspondance en table 1. Aux règles de la table 1 s'ajoute la représentation des diphtongues par l'association des 2 ou 3 phonèmes qui la constitue, séparés par '&' ("x" => /k&s/ par exemple). Ces règles permettent d'identifier les graphèmes porteurs de l'information phonétique (/_/ dans le cas des graphèmes muets), et sont utilisées pour explorer l'espace latent en sortie de l'encodeur comme décrit en section 4.2.

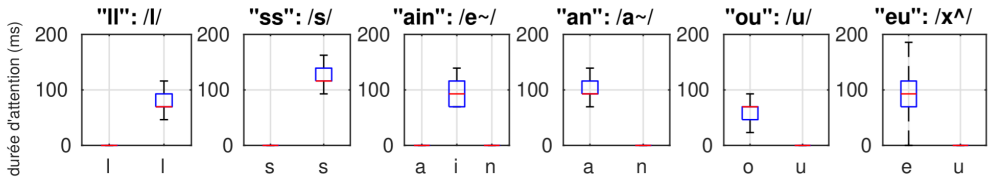


FIGURE 1 – Schémas des durées d'activation (en ms) de 6 phonèmes complexes.

Schémas	Activation	Exemples
C C	_ C	"nn", "ll", "ss"
V V	V _	"an", "ou", "au"
V V V	_ V _	"eau", "ain"

TABLE 1 – Règles d'activation des graphèmes. C et V représentent un caractère dans un phonème consonne et voyelle respectivement. _ représente un caractère muet.

4 Expériences et Résultats

4.1 Modèle et données

Le modèle Tacotron2 utilisé dans cette étude diffère légèrement de l'implémentation partagée par NVIDIA [1]. Comme (Lenglet *et al.*, 2021), notre modèle implémente la correction de Gate Loss, ainsi que la possibilité de présenter des entrées orthographiques et/ou phonétiques. Le décodeur génère 2 trames de spectrogramme à la fois. Nous avons observé que générer 2 trames à la fois permettait d'accélérer l'inférence, sans détériorer les performances du modèle. De plus, une tâche parallèle de prédiction phonétique à partir de la sortie de l'encodeur est mise en place pour les entrées orthographiques. Après mise en contexte par l'encodeur, une couche de projection linéaire associée à une fonction softmax réalise la classification des plongements orthographiques parmi l'ensemble des phones décrit par (Bailly *et al.*, 2021) [2], auquel s'ajoutent les diphtongues, pour un total de 63 phones. Pour établir une classification par graphème, le corpus a été annoté phonétiquement en suivant les règles d'activation observées en section 3.2. L'erreur d'entropie croisée de cette couche de classification est ajouté à l'erreur globale du modèle avant la rétro-propagation du gradient. Cette tâche parallèle permet d'aider à structurer l'espace latent lors de l'apprentissage et est évaluée en section 4.2.2.

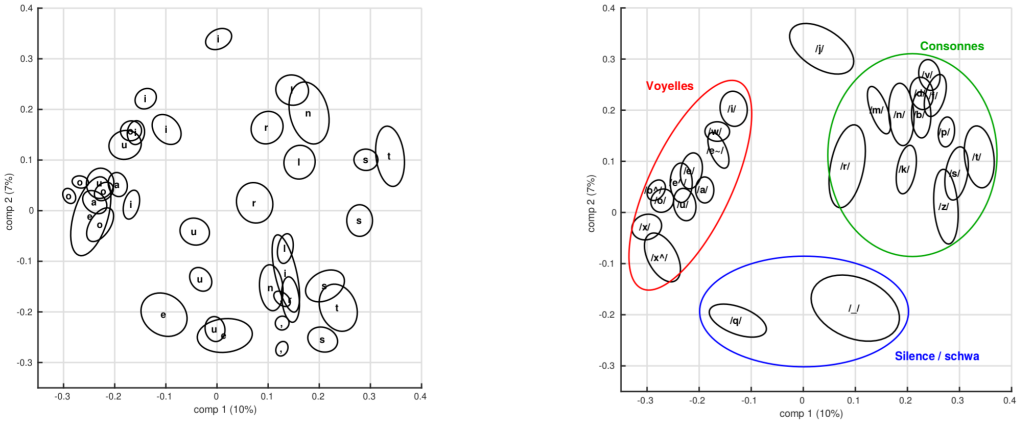
Ce modèle est entraîné sur la nouvelle segmentation d'une partie du corpus français M-AILABS proposée par (Bailly *et al.*, 2021). Nous avons sélectionné un ensemble de 29557 phrases tirées de 4 romans lus par Nadine Eckert-Boulet. Toutes les phrases sélectionnées sont présentes sous forme orthographique et phonétique. 5% de ce corpus est mis de côté pour le test, soit 1477 phrases. Le modèle est entraîné sur les 2 types d'entrées jusqu'à stabilisation de l'erreur sur la base de test, ce qui représente environ 100 époques. Pour les sections 4.2 et 4.3, les 1477 phrases du corpus de test sont ensuite synthétisées en utilisant l'entrée orthographique. Les plongements des graphèmes d'entrée, mis en contexte par l'encodeur sont enregistrés en parallèle de la synthèse, ainsi que la prédiction phonétique associée. Le vocodeur utilisé est Waveglow (Prenger *et al.*, 2019).

1. https ://github.com/NVIDIA/tacotron2
2. https ://zenodo.org/record/4580406#%23.YI_qIyaxXmE

4.2 Représentation phonétique des plongements de caractère

4.2.1 Visualisation de l'espace phonétique

Afin de visualiser l'espace latent, la distance cosinus est calculée entre toutes les paires de plongements du corpus de test. La matrice de distances ainsi obtenue est projetée en dimensions réduites par mise à l'échelle multidimensionnelle (MDS) (Kruskal, 1978). Les 2 premières composantes de cette projection sont données en figure 2, en considérant les plongements par leur graphème d'origine (2a) ou leur correspondance phonétique (2b), obtenue par la méthode décrite en section 3.2.



(a) Par graphème (15 graphèmes plus fréquents + ",") (b) Par phone (25 phones plus fréquents + /_/)

FIGURE 2 – MDS des plongements du corpus de test. Les ellipses montrent l'étalement de chaque groupe selon les 2 demi-axes principaux, avec une amplitude d'un écart-type.

Plusieurs groupes apparaissent pour chaque graphème, confirmant les résultats de (Perquin *et al.*, 2020). En revanche, la représentation par phonème fait apparaître des groupes uniques et distincts. De plus, on retrouve une proximité entre les plongements correspondant à des phonèmes de même type : les phonèmes de voyelles se distinguent des consonnes et des silences. Au sein même des consonnes, une séparation apparaît entre les consonnes plosives, nasales et fricatives. Le groupe des phonèmes silencieux /_/ regroupe la plus grande variété de graphèmes différents : tous les graphèmes muets tels que décrits en section 3.2, ainsi que les ponctuations et les espaces.

Ces observations suggèrent que l'encodeur de Tacotron2 apprend à représenter dans un espace phonétique les plongements de la séquence d'entrée, même si cette dernière est entièrement orthographique. Les ellipses de dispersion des groupes phonétiques suggèrent que le modèle est capable de générer des variations pour chaque phonème, basées sur le contexte dans lequel celui-ci se trouve. Ces ellipses sont plus restreintes pour les consonnes que les voyelles, ce qui est cohérent avec le phénomène de coarticulation qui impacte davantage les voyelles que les consonnes (Modarresi *et al.*, 2004). La structuration de cet espace latent relève donc d'un compromis entre 1) la distinction à faire entre les phonèmes en vue de leur prononciation et 2) laisser suffisamment de marge à chaque phonème pour permettre une coarticulation naturelle. L'évaluation du rapport entre les distances intra-classes et inter-classes, menée en amont de cette étude, montre que l'apprentissage conjoint des entrées orthographiques et phonétiques, ainsi que la tâche de prédiction phonétique à partir des entrées orthographiques, favorisent la désintrication des ellipses phonétiques. La proximité entre les phonèmes dont la prononciation est proche suggère que cet espace encode des informations acoustiques qui pourront être utilisées par le décodeur lors de l'inférence ; cette hypothèse sera explorée en section 4.3.

4.2.2 Évaluation de la prédiction phonétique

Les prédictions phonétiques obtenues par le classifieur à partir de la sortie de l'encodeur sont comparées aux transcriptions du corpus original par les règles de la section 3.2. La matrice de confusion du modèle est donnée en figure 3a.

Le modèle atteint une précision de 99% sur l'ensemble des phones. Ces performances sont supérieures à celles de (Perquin *et al.*, 2020) qui obtenait un taux d'erreur de 12.8% sur les phonèmes en entraînant un classifieur sur la sortie de l'encodeur d'un modèle Tacotron (Wang *et al.*, 2017). Ce gain peut s'expliquer par l'utilisation d'un modèle de synthèse vocale plus récent, entraîné sur des entrées à la fois orthographiques et phonétiques, et avec cette tâche supplémentaire de transcription 1 pour 1 entre graphèmes et phonèmes dès le début de son entraînement.

L'analyse plus détaillée de la matrice de confusion fait apparaître quelques erreurs fréquentes, résumées dans le tableau 2. Ces "erreurs" relèvent davantage d'un choix de style adopté par le locuteur que d'erreurs phonétiques. La majorité des distinctions entre la prédiction et le corpus d'origine apparaissent en fin de mot, dans le cadre de liaisons facultatives. L'écoute des synthèses concernées révèle que la prédiction du modèle est en accord avec la prononciation produite, mais que cette prononciation diffère de la prononciation adoptée dans le corpus original. Cette méthode de transcription automatique Graphème à Phonème par l'ajout d'une tâche supplémentaire lors de l'apprentissage d'un modèle de synthèse vocale permet donc d'associer deux avantages majeurs : 1) une excellente précision, et 2) une transcription adaptée à la synthèse correspondante, pertinente vis-à-vis des habitudes de style de parole du locuteur original (liaisons facultatives, schwas, etc..).

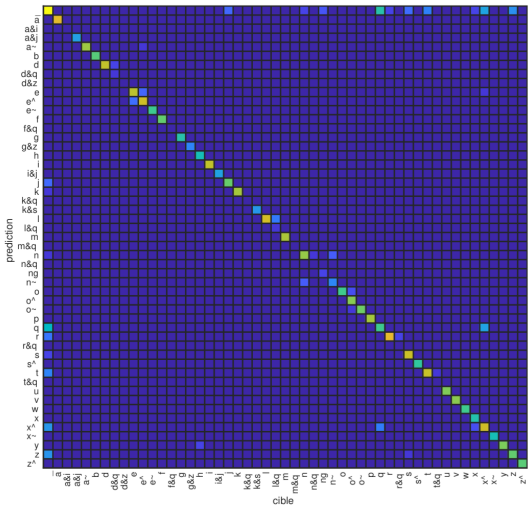
Phonèmes confondus		**Explications**	**Exemples**
/q/	/x^/	Schwas ou fin de mot appuyée	"quelqu**e**s rares fenêtres"
/o/	/o^/	Choix d'harmonisation vocalique	"**O**tons nos souliers"
/r/, /s/, /z/, /t/	/_/	Liaisons facultatives	"si tu n'es pas_heureux"

TABLE 2 – Confusions courantes révélées par la matrice de la figure 3a.

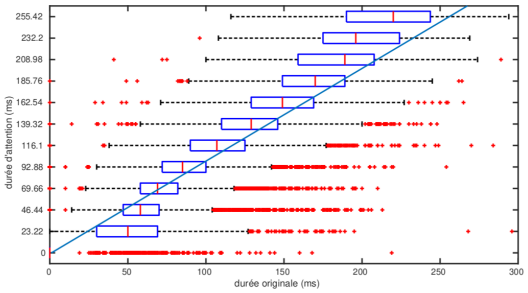
4.3 Analyse acoustique de l'espace des plongements

4.3.1 Segmentation automatique par l'attention

Afin d'identifier les paramètres acoustiques qui seraient encodés dans les plongements, le première étape consiste à isoler les portions du signal de sortie pendant lesquelles un plongement détermine la sortie audio. On utilise pour cela la procédure décrite en section 3.1. Afin de vérifier les performances de cette méthode, les durées d'activation ainsi calculées sont comparées aux durées des phonèmes dans le corpus initial obtenues par alignement semi-automatique. On adopte les règles définies en section 3.2 pour déterminer à quel plongement orthographique doit correspondre la durée du phonème : les caractères muets sont considérés comme ayant une durée de 0 secondes. Pour adopter le même rythme que la voix originale, les synthèses sont générées en prédiction, c'est-à-dire que les trames originales sont données au réseau en remplacement des trames calculées à chaque pas de temps du décodeur. Les résultats de cette comparaison sont donnés en figure 3b. La lecture de la carte d'attention permet de prédire les durées des phonèmes avec une corrélation de 0.88. La méthode proposée limite l'estimation des durées à un nombre pair de trames d'une longueur fixée par le modèle acoustique à 11.61ms (pour rappel : le modèle prédit 2 trames à la fois). Cette discrétisation de l'estimation peut expliquer une partie des erreurs de prédiction observées. Les performances de



(a) Matrice de confusion de la prédiction phonétique à partir de la sortie de l'encodeur. Tous les 80907 plongements du corpus de test sont pris en compte. Précision : 99%



(b) Comparaison entre les durées des phonèmes et la durée d'attention calculée suivant la méthode décrite en section 3.1.

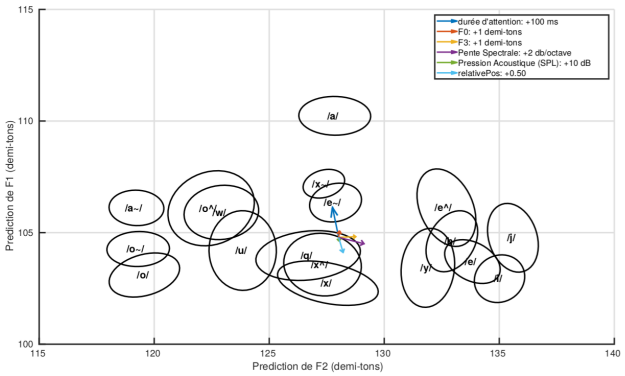
FIGURE 3 – Évaluation des méthodes proposées en section 3

cette méthode permettent d'envisager la segmentation automatique du corpus de test en vue d'une analyse acoustique.

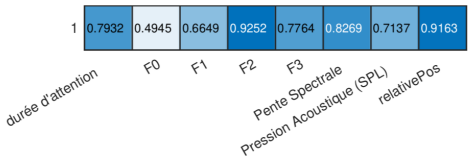
4.3.2 Corrélations acoustiques et contrôle

Suite aux observations de la section 4.3.1, la segmentation audio du corpus de test est réalisé en suivant la procédure décrite en section 3.1. On ne considère dans cette section que les plongements associés à des phonèmes voyelles, afin d'intégrer des mesures acoustiques dépendantes du voisement. 8 mesures sont considérées : la durée d'attention, F0, F1, F2, F3 la pente spectrale, le niveau de pression sonore, ainsi que la position relative du plongement dans la séquence d'entrée. Les mesures acoustiques sont effectuées automatiquement avec Praat (Boersma, 2001). Les plongements sont projetés dans un espace de dimensions réduites par MDS comme détaillé en section 4.2.1. Une régression multi-linéaire par paramètre est réalisée afin d'obtenir une estimation de ces paramètres en fonction de la position du plongement dans la MDS. Les coefficients de corrélation de ces régressions sont donnés en figure 4b. Pour représenter l'espace latent, on affiche en figure 4a les plongements de voyelles en fonction de l'estimation de leurs premier et deuxième formants. On retrouve en partie le triangle vocalique formé par le trinôme /a/, /i/ et /u/. Les voyelles ouvertes /o^/ et /e^/ sont bien placées entre les versions mi-fermées /o/ et /e/ et la voyelle la plus ouverte /a/. La prédiction des formants semble moins précise sur les phonèmes les moins fréquents dans le corpus : les nasales /a~/, /o~/, /x~/ et /e~/, ainsi que /o/.

Les régressions par paramètre permettent d'envisager l'espace des plongements comme un espace de contrôle effectif : chaque paramètre peut être modifié en déplaçant les plongements dans la direction entraînant le maximum de variation de ce paramètre. A titre d'exemple, le déplacement induit par une modification arbitraire des paramètres à l'étude, est affiché en figure 4a. Le contrôle de la synthèse par un biais appliqué dans cet espace profite des covariations apprises par le modèle. On note par exemple qu'augmenter la durée portée par l'attention aux plongements de 100ms est



(a) Gradients de variation des paramètres acoustiques



(b) Coefficients de corrélation entre les paramètres acoustiques mesurés et leur prédiction par régression multi-linéaire sur leurs coordonnées dans la MDS.

FIGURE 4 – Visualisation des paramètres acoustiques dans l'espace latent réduit par MDS

accompagné d'une augmentation du premier formant de 1.2 demi-tons, cohérente avec une durée plus importante des voyelles ouvertes (O'Shaughnessy, 1981). De même, une légère augmentation de F1 en augmentant F0 rejoint la simulation d'un effort vocal plus important (Liénard & Di Benedetto, 1999). À l'inverse, l'augmentation de la position relative du phonème est associée à une réduction de F1 qui mimique la réduction de l'ouverture des voyelles en fin de phrases. L'amplitude relativement faible des déplacements dans le plan de la figure 4a est rassurant vis-à-vis des possibilités de contrôle dans cet espace ; le spectre des phonèmes ne doit pas être dénaturé au point d'être confondu avec une autre voyelle lors de la modification d'un paramètre de durée par exemple.

La MDS étant un transformation linéaire de l'espace, l'inverse de la matrice de passage permet de projeter cette translation dans l'espace des plongements. Le vecteur résultant peut ensuite être ajouté à tous les plongements en sortie de l'encodeur avant le passage dans le décodeur, afin de modifier une ou plusieurs caractéristiques de la voix. L'évaluation du contrôle de la synthèse par cette méthode est encore à l'étude.

5 Conclusions

Cette article présente une méthode d'analyse acoustique et phonétique des représentations internes d'un modèle de synthèse vocale à l'état de l'art. Cette étude a montré que la séquence de graphèmes donnée en consigne au modèle de synthèse était mise en contexte par l'encodeur pour en calculer une représentation phonétique. De plus, cette analyse montre que les représentations phonétiques calculées par l'encodeur contiennent non seulement les cibles acoustiques à atteindre, mais également des informations de rythme et de positionnement dans la phrase. La localisation des paramètres acoustiques dans les plongements permet d'imaginer le contrôle de ces derniers, par exemple par l'ajout d'un biais global sur la phrase à synthétiser qui déplacerait tous les plongements dans une direction correspondant à la variation du paramètre choisi. Ce type de contrôle, ainsi qu'une analyse approfondie des indices supra-segmentaux de rythme et de position fera l'objet de futures recherches.

Remerciements

Ces recherches sont financées par la BPI dans le cadre du projet THERADIA et par MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). Ces travaux ont l'accès à HPC/IDRIS sous l'attribution 2021-AD011011542R1 faite par GENCI.

Références

BAHDANAU D., CHO K. & BENGIO Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv :1409.0473*.

BAILLY G., PERROTIN O. & LENGLET M. (2021). Ressources for End-to-End French Text-to-Speech Blizzard challenge.

BOERSMA P. (2001). Praat, a system for doing phonetics by computer. *Glot. Int.*, **5**(9), 341–345.

BOSSE M.-L. & VALDOIS S. (2009). Influence of the visual attention span on child reading performance : a cross-sectional study. *Journal of research in reading*, **32**(2), 230–253.

BURKART N. & HUBER M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, **70**, 245–317.

HSU W.-N., ZHANG Y., WEISS R. J., CHUNG Y.-A., WANG Y., WU Y. & GLASS J. (2019). Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization. In *ICASSP*, p. 5901–5905 : IEEE.

KRUSKAL J. B. (1978). *Multidimensional scaling*. Number 11. Sage.

LENGLET M., PERROTIN O. & BAILLY G. (2021). Impact of segmentation and annotation in french end-to-end synthesis. In *11th ISCA Speech Synthesis Workshop*, p. 13–18 : ISCA.

LIÉNARD J.-S. & DI BENEDETTO M.-G. (1999). Effect of vocal effort on spectral properties of vowels. *The Journal of the Acoustical Society of America*, **106**(1), 411–422.

MODARRESI G., SUSSMAN H., LINDBLOM B. & BURLINGAME E. (2004). An acoustic analysis of the bidirectionality of coarticulation in vcv utterances. *Journal of Phonetics*, **32**(3), 291–312.

O'SHAUGHNESSY D. (1981). A study of french vowel and consonant durations. *Journal of Phonetics*, **9**(4), 385–406.

PERQUIN A., COOPER E. & YAMAGISHI J. (2020). An investigation of the relation between grapheme embeddings and pronunciation for tacotron-based systems. *arXiv preprint arXiv :2010.10694*.

PRENGER R., VALLE R. & CATANZARO B. (2019). Waveglow : A flow-based generative network for speech synthesis. In *ICASSP*, p. 3617–3621 : IEEE.

REN Y., HU C., TAN X., QIN T., ZHAO S., ZHAO Z. & LIU T.-Y. (2020). Fastspeech 2 : Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv :2006.04558*.

SHEN J., PANG R., WEISS R. J., SCHUSTER M., JAITLY N., YANG Z., CHEN Z., ZHANG Y., WANG Y., SKERRV-RYAN R. *et al.* (2018). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*, p. 4779–4783 : IEEE.

SKERRY-RYAN R., BATTENBERG E., XIAO Y., WANG Y., STANTON D., SHOR J., WEISS R. J., CLARK R. & SAUROUS R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv preprint arXiv :1803.09047*.

TACHIBANA H., UENOYAMA K. & AIHARA S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *ICASSP*, p. 4784–4788.

TITS N., WANG F., HADDAD K. E., PAGEL V. & DUTOIT T. (2019). Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis. *arXiv preprint arXiv :1903.11570*.

WANG Y., SKERRY-RYAN R., STANTON D., WU Y., WEISS R. J., JAITLY N., YANG Z., XIAO Y., CHEN Z., BENGIO S., LE Q., AGIOMYRGIANNAKIS Y., CLARK R. & SAUROUS R. A. (2017). Tacotron : Towards end-to-end speech synthesis. In *Proceedings of Interspeech*, p. 4006–4010, Stockholm, Sweden.

Interspeech 2022
18-22 September 2022, Incheon, Korea

Speaking Rate Control of end-to-end TTS Models by Direct Manipulation of the Encoder's Output Embeddings

Martin Lenglet, Olivier Perrotin, Gérard Bailly

Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, France
{martin.lenglet,olivier.perrotin,gerard.bailly}@grenoble-inp.fr

Abstract

Since neural Text-To-Speech models have achieved such high standards in terms of naturalness, the main focus of the field has gradually shifted to gaining more control over the expressiveness of the synthetic voices. One of these leverages is the control of the speaking rate that has become harder for a human operator to control since the introduction of neural attention networks to model speech dynamics. While numerous models have reintroduced an explicit duration control (ex: FastSpeech2), these models generally rely on additional tasks to complete during their training. In this paper, we show how an acoustic analysis of the internal embeddings delivered by the encoder of an unsupervised end-to-end TTS Tacotron2 model is enough to identify and control some acoustic parameters of interest. Specifically, we compare this speaking rate control with the duration control offered by a supervised FastSpeech2 model. Experimental results show that the control provided by embeddings reproduces a behaviour closer to natural speech data.
Index Terms: speech synthesis, embeddings analysis, natural control, duration control

1. Introduction

Deep neural Text-To-Speech systems such as Tacotron [1, 2] or FastSpeech [3], combined with neural vocoders like WaveNet [4], WaveRNN [5] or WaveGlow [6] produce more realistic voices than ever. As a result, numerous studies now focus on the rendition of the expressiveness [7, 8], whose control remains a ongoing challenge. In particular, prosody is known to convey co-verbal information that is desirable to make the interaction with a synthetic voice as natural as possible [9]. The accurate manipulation of prosodic parameters of interest such as pitch, energy or speaking rate is therefore a requirement for an interactive TTS system.

One approach to enable the control of these parameters at inference time consists in adding layers to the model in order to learn how to explicitly retrieve this information from the input sequence [3, 10]. Doing so, this information can be modified before being reintroduced into the decoding layer, resulting in a finer control of the output prosody. While this method enables an independent control of these parameters, it requires various preprocessing to extract alignments and acoustic parameters beforehand. Additionally, the proposed independent control may not correspond to the natural behaviour of the voice.

An alternative is to use implicit representation to bias the model toward the desired prosody [7, 11]. Via a so-called prosodic or reference encoder of the target speech signals, style and speaker embeddings model residual loss not yet explained by text input. During inference, a target prosodic example can then be used to complement the input text. While control of style may capture subtle natural co-variations, the semantics of control parameters is often given a posteriori.

In this paper, we introduce a new control for end-to-end TTS models: *Embedding Bias*. By analysing the phonetic embeddings at the encoder's output, we identify acoustic and paralinguistic parameters that are encoded in these latent representations, as well as their co-variations with other phonetic dimensions, learnt from the training data. We show how this information can be used to bias phonetic embeddings in order to control the speaking rate of the model, without the need for any additional data during the training phase. We implement and investigate this duration control on the embedding spaces of both Tacotron2 and FastSpeech2 models, whose biased embeddings are then fed to their attention mechanism and duration predictor, respectively. We compare this control to the explicit duration control provided by FastSpeech2.

2. Related Work

The explicit prediction of low-level prosodic parameters such as F_0, duration and energy from the embedding space of encoder-decoder TTS models has led to excellent performance in disentangling these parameters [3, 10] at the expense of preserving the natural co-variations between them. Moreover, duration control usually applies a uniform gain to all phones, whereas variations of phone duration with speaking rate depends on its phonemic class and position in the sentence [12]. Whether the loss of both supra-segmental acoustic co-variations and non-linear duration variations at a segmental level degrades naturalness is still an open question and is investigated here. The opposite direction that consists in biasing the encoder output with an implicit representation of an audio sample learnt by a reference encoder (Global Style Tokens [7, 11, 13], Variational Auto Encoders [14, 15] or speaker encoders [16]) supposedly better preserves the co-variations of prosodic parameters. However, if most implementations allow to successfully identify dimensions in the obtained latent space to control low-level prosodic parameters, few quantitative studies had statistically analysed variations and co-variations of prosodic parameters introduced by an implicit control both at segmental and supra-segmental levels. Also, methods for systematic analyses of latent spaces are rarely given, with exceptions such as [17] who performed an *a posteriori* analysis using a crowd-sourced subjective evaluation of synthesis.

The difference between concatenation or addition of the style and text encoders outputs is not well described in the literature, yet the addition intuitively corresponds with a translation in the embedding space. Therefore, can we derive the appropriate translation for a given prosodic parameter modification from an analysis of the embedding space, without the need to train a reference encoder? Previous work on embedding space analysis showed promising results in terms of phonetic [18] and acoustic [19] structuring of the embedding space, but no control were yet identified from these analyses.

10.21437/Interspeech.2022-759

3. Proposed Method

We aim at performing an acoustic analysis of the latent space outputted by the encoder of an end-to-end TTS model, and use this analysis to exhibit an embedding bias that can monitor the speaking rate of the model. This method could be applied to any encoder-decoder architecture which uses an attention mechanism or a duration predictor. Both cases are implemented, taking Tacotron2 [2] and FastSpeech2 [3] as examples.

3.1. Encoder-decoder TTS models

Our implementation of Tacotron2 (*TC*) builds on the one shared by NVIDIA [20]. Following [21], *TC* uses a Gate Loss correction and is trained on both orthographic and phonetic transcripts, which are known to benefit to both types of inputs [22]. Additionally, the decoder generates two mel-spectrogram frames per step. Empirical analysis showed that generating 2 frames at a time did not degrade the overall quality of the synthetic speech, while speeding the inference process. FastSpeech2 (*FS*) strictly follows an early implementation [23]: the pitch predictor is trained on F_0 values instead of continuous wavelet transform in later versions. A Tacotron2-type post-net is added after the decoder. Also, pitch and energy values are averaged per phone instead of per frames, and normalised.

Both *TC* and *FS* are trained on a subset of the new segmentation of the French M-AILABS dataset provided by [24]. This subset includes 29557 utterances (more than 25h) of audiobook recordings from four novels uttered by Nadine Eckert-Boulet (NEB). 5% of this corpus (1477 utterances) was randomly picked as the test set. This dataset provides both orthographic and phonetic transcripts for every utterance. Only the phonetic transcripts (together with spaces and punctuation when associated with pauses) were used for *FS*, which was also provided a hand-checked phonetic alignment to train its duration predictor. Both models were trained until convergence, which took about 100 epochs. The post-net is bypassed during the first 10 epochs, while the learning rate is fixed at 10^{-3}. After this startup, the learning rate decreases exponentially until reaching 10^{-5} after 90 epochs. The batch size is set to 32 for both models. The vocoder used is WaveGlow [6].

3.2. Identification of Acoustic Parameters in Embeddings

After training, the entire test set was synthesised with both models, using the phonetic input. Together with the usual audio output, embeddings computed by the encoder of both models are saved for acoustic analysis, as well as the attention map from *TC* and the duration predictions from *FS*.

3.2.1. Automatic Segmentation of the Synthesised Audio Signal

In *TC*, durations of input phones are computed using the durations of their respective activations in the attention map [25]. The duration of output phones predicted with this method were compared to Ground-Truth phones duration. Syntheses were produced using teacher-forcing to ensure the same dynamic as the Ground-Truth. We measured a correlation of 0.88 on phones (durations of silences from punctuation marks are excluded), which made us consider this method for large scale acoustic analysis. Segmentation in *FS* is straightforward, the duration predictor providing the number of frames for each phone. An acoustic analysis of each phone is performed with Praat [26]. Several acoustic parameters are considered: phone duration, fundamental frequency (F_0), first three formants (F_1, F_2, and F_3), and energy (Sound Pressure Level).

Table 1: *Correlation coefficients between acoustic features predicted from MDS coordinates and measured on synthesis. log(d) = logarithm of the duration ; E = Energy.*

Model	**log(d)**	**F_0**	**F_1**	**F_2**	**F_3**	**E**
Tacotron2	0.83	0.51	0.70	0.93	0.75	0.67
FastSpeech2	0.89	0.86	0.84	0.91	0.74	0.87

3.2.2. Acoustic Analysis of the Embedding Space

The synthesis of the entire test set provides a total of 51746 phone embeddings that encode contextual information introduced by the encoder of each model. To consider the voiced-dependent acoustic features (section 3.2.1), only the 22528 vowels of the test set are studied in this section. The relationship between embeddings and acoustic features measured on the corresponding synthesised audio segments is derived as follows: 1) Dimensional reduction of the embedding space with Multi-dimensional Scaling (MDS) [27]. A distance matrix between embeddings is first calculated using cosine distance. 2) A projection matrix is derived to enable transitions between the initial embedding space and the reduced MDS space. 3) All the acoustic features are individually approximated by least square multilinear regression from embeddings coordinates in the MDS. This procedure is similar to [19], but is applied on phone embeddings instead of utterance-wise style embeddings.

The approximation of acoustic parameters from the MDS coordinates is compared to the measured acoustic features on the synthesised signals and correlation coefficients are shown in table 1. Phone durations are computed in logarithmic scale, because this gave better correlations. Same goes for F_0, F_1, F_2 and F_3 which are expressed in semitones for better approximations. These correlations indicate that most of these acoustic features are well encoded in the embeddings. Note that a lower correlation does not mean that the model does not implement this acoustic feature, but rather that this feature is not encoded in the phone embeddings alone (note that duration, F0 and energy encoders of *FS* further contextualise embeddings with CNNs) or not correlated in a linear way. As a result, this feature is less likely to be easily controllable by modifying the embeddings before passing through the decoder. On the contrary, high correlations emphasise the features that are encoded in this latent space: phone duration is well encoded by every model, as well as spectral clues such as formants. *FS* has better correlations of prosodic measurements like F_0 and energy, which are trained to be predicted by the model from the very same embeddings.

3.3. Acoustic control

From the regressions described in section 3.2.2, the gradient of each acoustic feature in the MDS is computed. This vector, called *embedding bias*, is the leverage used to control one particular feature at a time: a translation along this vector is added to all the embeddings of an utterance before passing through the decoder. The regression is used to evaluate the magnitude of translation needed to induce the desired modification of the acoustic feature. This study specifically evaluates the control given by the duration embedding bias, expressed in log-duration. Hence the addition of a bias in the log domain is equivalent to applying a multiplying factor on phone duration. We empirically identified that a correcting factor k was needed to achieve the desired modification of phone duration, resulting in a corrected translation of $k * log(m)$ to multiply phone duration by m. $k = 2.94$ and $k = 2.33$ for *TC* and *FS* respectively.

In the case of *FS*, the embedding bias is applied before duration prediction, and predicted duration from the biased embeddings is used for decoding, without any external input. We showed in a preliminary study that an embedding bias computed on vowel embeddings alone is more efficient in inferring duration modification in the synthesis signal, supported by the fact that vowels duration show more variability than consonants [12]. In the following, the bias is derived from the vowel embeddings space but applied on all input phone embeddings at inference.

4. Experiments and Results

4.1. Models and test set

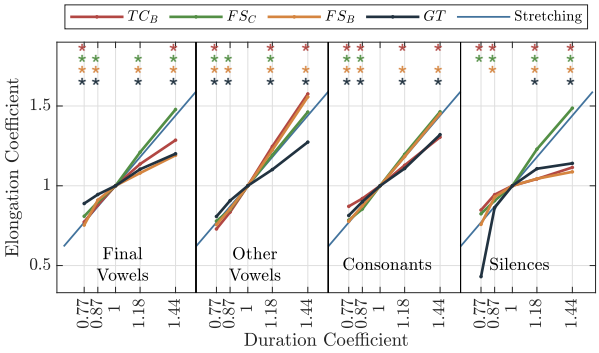
In this section we will investigate and compare the efficiency of the embedding bias control on *TC* and *FS*. In addition, two baselines are added: *FS* with explicit duration control (without embedding bias) and a simple linear time-interpolation of the mel-spectrogram output of an unbiased *TC* (resp. *FS*) to change the full duration of the signal before feeding it to the neural vocoder. In both baselines, a similar modification of duration is applied on all phones, but *FS* has the chance to make some acoustic modifications through the decoding process. In the following, TC_B, FS_B, FS_C and *stretching* refer to *TC* with embedding bias, *FS* with embedding bias, *FS* with explicit duration control, and mel-spectrogram interpolation, respectively.

The test set described in section 3.1 is synthesised with 4 duration coefficients, chosen to be representative of the phone rate distribution of the training dataset. These coefficients $m_i = \{0.77, 0.87, 1.18, 1.44\}$ are chosen to reach $i = \{+2, +1, -1, -2\}$ standard deviation around the mean phone rate, respectively.

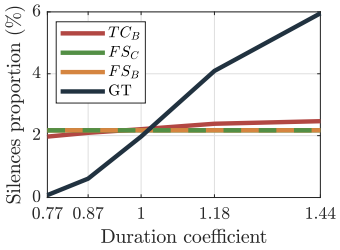
4.2. Non-linear duration modification

For each synthesised signal with a given duration coefficient, the duration of each phone is measured (see section 3.2.1), and divided by the mean duration of its phone class synthesised with the same model without duration control, to provide an elongation coefficient. Fig. 1a displays the average elongation coefficient per duration coefficient, model, and phone class. Final vowels are vowels just preceding a silence in the audio signal. For each phone class, the diagonal corresponds to the *stretching* condition, where the elongation coefficient equals the duration coefficient. The red, green and yellow curves correspond to TC_B, FS_C, FS_B, respectively. Moreover, average phone elongation coefficients were also calculated on the ground truth train database (*GT*) and reported in dark blue. A Kruskal-Wallis rank-sum test performed on the per-phone elongation coefficients showed a significant effect of both phone class and duration control ($p < 10^{-3}$). A post-hoc Wilcoxon rank-sum test then assessed for each phone class and duration coefficient whether each method significantly differs from the *stretching* conditions. Significance ($p < 10^{-3}$) is displayed by coloured stars above each data point. Fig. 1b shows the ratio between the number of pauses longer than 30 ms in the audio signal and the number of phones in the text input for each duration coefficient on TC_B, FS_C, FS_B and *GT* (by nature, this ratio do not vary with duration control for FS_C, FS_B and *stretching*).

Concerning elongation coefficients (Fig. 1a), FS_C follows the diagonal: as expected, frames are linearly duplicated through duration control for any class of phonemes. On the contrary, *GT* data displays non-linear behaviours that are consistent with [12] findings. These behaviours are partly followed by the embedding bias-controlled model. Looking first at slower



(a) *Elongation coefficient is the mean phone elongation compared to the unbiased voice. * indicates a significant difference with stretching.*



(b) *Silences proportion is the ratio between the number of silences in the audio signal and number of phones in the text input.*

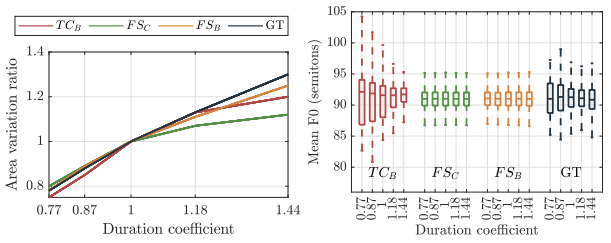
Figure 1: *Impact of duration control for each model and GT.*

speaking rates ($m_i > 1$), *GT* displays a saturation for final vowels and silences whose mean durations are large for average speaking rate (125 ms and 213 ms, respectively) and weakly lengthened as the speaking rate decreases. This behaviour has been learnt by TC_B and FS_B. Regarding other vowels and all consonants, *GT* shows a linear lengthening with duration control but to a lesser extent than *stretching*. This is compensated by the introduction of pauses in the *GT* signals: Fig. 1b displays three times more pauses in *GT* when the speaking rate is 1.44 times slower. Conversely, FS_B is unable to add any pauses in the signal, and the effect is negligible for TC_B. Alternatively, both models compensate by expanding the vowels longer than the stretch (Fig 1a). On consonants, TC_B seems to have learnt *GT* behaviour, wile FS_B follows the *stretching* trend. Looking now at higher speaker rates ($m_i < 1$), *GT* final vowels are preserved while silences are dramatically shortened or deleted (Fig. 1b). This behaviour was not replicated by any model. For other vowels and consonants, *GT* and all models follow a linear shortening of phones matching *stretching*.

Globally, *GT* duration modification is mainly performed with pauses addition and deletion, that are hardly managed by the embedding bias-controlled models. Regarding the observed non-linearity per class of phonemes, TC_B follows at best the *GT* behaviours, even though it compensates for the lack of pause addition by vowel lengthening. Both TC_B and FS_B follows the saturation of final vowels and pauses that are imposed by the data distribution, but FS_B mainly follows the *stretching* behaviour otherwise, showing that *TC* better models non-linearities in duration modification than *FS*, when using a similar embedding-bias control policy.

4.3. Co-variations of acoustic parameters

To investigate the co-variations of acoustic parameters with duration control, we first derived F_1 and F_2 values for all /a,i,u/



(a) *Vocalic triangle area.* (b) *Fundamental frequency.*

Figure 2: *Acoustic parameter variations by model and by speed.*

vowels present in the synthesised signals with the different models and duration coefficients. We then derive the area between the three vowels on the F_1-F_2 plane. The ratios between the area obtained for each duration coefficient and without duration modification are reported in Fig. 2a for each model and *GT*. For higher speaking rate, a similar linear compression of the vocalic triangle is observed for all models and *GT*, typical of an undershooting of the vowel targets. For lower speaking rates, *GT* displays an expansion of the vocalic triangle, which is successfully replicated by FS_B and TC_B, with a slight saturation for the 1.44 coefficient. FS_C shows a stronger saturation.

Fig. 2b shows mean utterance F_0 values per model and duration coefficient. *GT* data shows an increase in F_0 median and variability with highest speaking rates, which are well replicated by TC_B. Conversely, none of the *FS* models display any co-variation of F_0 with duration control. Overall, co-variations of features learnt in an unsupervised way, like formants, are well replicated by both models, while the the F_0 and duration prediction tasks implemented in *FS* lead the latter to ignore the co-variations between those parameters.

4.4. Listening Experiment

To investigate the effect of segmental and supra-segmental variations and co-variations of prosodic parameters on perception, we conducted a listening experiment where each model was evaluated against the *stretching* method. A CMOS protocol was followed [28], where participants were presented a (model,*stretching*) pair and asked which of these voice speed renderings felt the most natural. Each pair consisted of one sentence synthesised with one of the three models (FS_C, FS_B, TC_B) and one of the four duration coefficients (0.77, 0.87, 1.18, 1.44) against its *stretching* counterpart. Order of presentation was counterbalanced. In total, 3 models $\times$ 4 duration coefficients $\times$ 18 sentences $\times$ 2 order of presentation = 432 pairs were evaluated. 42 participants recruited on Prolific [29] took part in the experiment, and each evaluated 72 stimuli following a Latin Square design so that every model, duration and sentence was equally seen by each subject. Table 2 reports the averaged CMOS obtained for each model and duration coefficient. A positive value indicates that the model was preferred over *stretching*, and conversely. A non-parametric Kruskal-Wallis test showed a significant effect of both duration control and models on the CMOS ($p < 0.001$). Post-hoc Wilcoxon tests by pairs were applied and a star in the Table indicates that the model shows a statistically different CMOS than the other two models for this duration coefficient ($p < 0.001$).

Table 2: *CMOS of duration control methods against stretching of unbiased synthesis from the same model.*

Model	0.77	0.87	1.18	1.44
TC_B	**-0.818***	**-0.544***	**0.525***	-0.004
FS_C	-0.079	0.048	**0.171***	**0.623***
FS_B	-0.075	-0.048	**-0.175***	-0.159

Overall, FS_B was considered as similar as *stretching* while TC_B shows more contrasting results, supporting that subjects were sensitive to the segmental and supra-segmental co-variations that are globally better modelled by TC_B. For higher speaking rates, TC_B was significantly less preferred than *stretching*. The prosodic parameters analysis highlighted a difference in F_0 variability between models for higher speaking rate may explain this failure. A further analysis of the training set showed that highest speaking rates often correspond to the expressive reading of dialogs between characters. Without any residual encoder to segment this paralinguistic information apart from text input, *TC* may have learnt an averaged representation of these characters, resulting in an unnatural speech depreciated by participants. By contrast, TC_B is preferred to *stretching* with the 1.18 duration coefficient. With this coefficient, the main difference between models lays in the non-linearity of phone duration (Fig. 1a), where TC_B closely matches the behaviour of *GT*. This is a case where the learning of co-variation is in favour of naturalness. Reaching the 1.44 duration coefficient, both embedding bias-controlled models are equally rated as *stretching*, while FS_C is preferred. We showed that at this speaking rate the addition of pauses in the signal is essential to prevent the over-lengthening of vowels sounds observed for TC_B and FS_B that could have been perceived as unnatural. Conversely, even though FS_C cannot add supplementary pauses, it has the ability to lengthen them to a greater extent. The preference of FS_C over *stretching* could also be due to a better conservation of phone transitions, that is yet to be verified.

5. Conclusions and Discussion

We proposed a method for the analysis of the embedding space of an encoder-decoder TTS model to derive an embedding bias that is applied to control a given prosodic parameter. It aims at 1) explicitly targeting a specific prosodic parameter, in opposition to reference encoders; 2) preserve the segmental and supra-segmental variations and co-variations in speech, contrary to learnt prosodic control models. Evaluation was performed on the control of speaking rate on both attention-based (*TC*) and duration predictor based (*FS*) methods. Objective analyses showed that while the prosodic parameters estimation implemented in *FS* cleared its embedding space of most of their corresponding segmental and supra-segmental co-variations, *TC* successfully modelled this information, and this was well perceived in a listening test. The possibility to add or remove pauses while modifying the speaking rate appears essential in order to model the natural behaviour of speech. Models that use explicit phonetic inputs (ex: *FS*) negate this phenomenon. Future works should elaborate on how to give this degree of freedom to synthesis models. This multi-dimensional segmental and supra-segmental prosodic parameter variations introduced by the embedding bias control invites to propose more feature-centred evaluations in the future, in conjunction with the control of other prosodic parameters.

6. Acknowledgements

This research has received funding from the BPI project THERADIA and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). This work was granted access to HPC/IDRIS under the allocation 2021-AD011011542R1 made by GENCI.

7. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proceedings of Interspeech*, Stockholm, Sweden, August 21-24 2017, pp. 4006–4010. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1452

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[3] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[4] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[6] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*. IEEE, 2019, pp. 3617–3621.

[7] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[8] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 595–602.

[9] D. Potdevin, "Vers des agents conversationnels animés sociaux: Quelle influence de l'intimité virtuelle sur l'expérience utilisateur et la relation-client?" Ph.D. dissertation, Université Paris-Saclay, 2020.

[10] D. S. R. Mohan, V. Hu, T. H. Teh, A. Torresquintero, C. G. Wallis, M. Staib, L. Foglianti, J. Gao, and S. King, "Ctrl-P: Temporal Control of Prosodic Variation for Speech Synthesis," in *Proc. Interspeech 2021*, 2021, pp. 3875–3879.

[11] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[12] N. Campbell, "Multi-level timing in speech," Ph.D. dissertation, Sussex University, U.K. Department of Experimental Psychology, 1992.

[13] M. Kim, S. J. Cheon, B. J. Choi, J. J. Kim, and N. S. Kim, "Expressive Text-to-Speech Using Style Tag," in *Proc. Interspeech 2021*, 2021, pp. 4663–4667.

[14] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6945–6949.

[15] W.-N. Hsu, Y. Zhang, R. J. Weiss, Y.-A. Chung, Y. Wang, Y. Wu, and J. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *ICASSP*. IEEE, 2019, pp. 5901–5905.

[16] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[17] P. van Rijn, S. Mertes, D. Schiller, P. M. Harrison, P. Larrouy-Maestri, E. André, and N. Jacoby, "Exploring Emotional Prototypes in a High Dimensional TTS Latent Space," in *Proc. Interspeech 2021*, 2021, pp. 3870–3874.

[18] A. Perquin, E. Cooper, and J. Yamagishi, "An investigation of the relation between grapheme embeddings and pronunciation for tacotron-based systems," *arXiv preprint arXiv:2010.10694*, 2020.

[19] N. Tits, F. Wang, K. E. Haddad, V. Pagel, and T. Dutoit, "Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis," *arXiv preprint arXiv:1903.11570*, 2019.

[20] NVIDIA, "Tacotron2 implementation." [Online]. Available: https://github.com/NVIDIA/tacotron2

[21] M. Lenglet, O. Perrotin, and G. Bailly, "Impact of segmentation and annotation in french end-to-end synthesis," in *11th ISCA Speech Synthesis Workshop*. ISCA, 2021, pp. 13–18.

[22] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.

[23] C.-M. Chien, "FastSpeech2 implementation." [Online]. Available: https://github.com/ming024/FastSpeech2

[24] G. Bailly, O. Perrotin, and M. Lenglet, "Ressources for End-to-End French Text-to-Speech Blizzard challenge," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4580406

[25] M. Lenglet, O. Perrotin, and G. Bailly, "Modélisation de la parole avec tacotron2 : Analyse acoustique et phonétique des plongements de caractère," in *Actes des Journées d'Etudes sur la Parole (JEP)*, Noirmoutiers, France, June 13-17 2022.

[26] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot. Int.*, vol. 5, no. 9, pp. 341–345, 2001.

[27] J. B. Kruskal, *Multidimensional scaling*. Sage, 1978, no. 11.

[28] I. T. Union, "Methods for objective and subjective assessment of quality," International Telecommunication Union, Tech. Rep. ITU-T P.800, 1998.

[29] S. Palan and C. Schitter, "Prolific.ac—a subject pool for online experiments," *Journal of Behavioral and Experimental Finance*, vol. 17, pp. 22–27, 2018.

The GIPSA-Lab Text-To-Speech System for the Blizzard Challenge 2023

Martin Lenglet, Olivier Perrotin, Gérard Bailly

Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Lab, France

{martin.lenglet,olivier.perrotin,gerard.bailly}@grenoble-inp.fr

Abstract

This paper describes the GIPSA-Lab submission to the Blizzard Challenge 2023. The Text-To-Speech system trained for this challenge is a Transformer-based non-autoregressive encoder-decoder architecture based on FastSpeech2. Updates of the FastSpeech2 framework were provided to specifically train the model on orthographic inputs, which is our main focus for this edition of the challenge. This model was trained with both orthographic and phonetic transcriptions of the same dataset. An additional phonetic prediction layer was added to the model. This additional layer enables to train the text encoder on phonetic prediction alone, without the need for audio recordings.

Index Terms: speech synthesis, mixed-inputs TTS, phonetic prediction

1. Introduction

Latest neural networks breakthroughs have largely improved performances of various automatic speech processing tasks, including Text-To-Speech (TTS). Latest neural TTS [1, 2, 3, 4], combined with neural vocoders [5, 6, 7] generate synthetic voices that closely mimic natural speech. However, the evaluation of synthetic speech naturalness is mostly conducted in favorable environments, on test stimuli which are very close to the training corpus. Thus, the good performances shown by neural TTS models may be overestimated compared to real-life applications.

The Blizzard Challenge 2023 aims at evaluating latest neural TTS systems in more challenging environments. More specifically, the Hub-Task of this challenge includes the evaluation of intelligibility of semantically unpredictable sentences and heterophonic homographs. The Spoke-Task on the other hand is a speaker adaptation task on a smaller dataset shared by the Blizzard organizers. Only orthographic sequences can be used as inputs in the submitted systems.

Our approach to this challenge is to propose a TTS system very close to the state-of-the-art model FastSpeech2 [4] but with the addition of phonetic prediction sub-task. FastSpeech2 is a fully parallel Transformer-based [8] architecture which implements 3 secondary tasks on top of the spectrogram prediction: pitch, energy and duration prediction. The duration prediction is the key factor of this parallel architecture, since it is necessary to realize the phone-to-frames alignment at the interface between the text encoder and the audio-decoder. However, this duration predictor is also the limiting factor to train FastSpeech2 on orthographic inputs, since the time-segmentation of the training set, necessary to train this predictor, is unclear when processing orthographic sequences. In this paper, we show how the letter-to-sound alignment proposed by Lenglet et al. [9] can be used to assign duration to the orthographic sequences in order to train a FastSpeech2 model on orthographic inputs. Moreover, we show that the addition of a phonetic prediction task from the output of the FastSpeech2 text encoder allows to train the model on <orthography|phonetic> pairs without the need for audio recordings. This setup helps learning phonetic transcriptions for words and contexts that are otherwise rarely found in classical audiobooks training corpora. We show through Blizzard results that this addition helped modelling heterophonic homographs. Results also show that our model is perceived as more natural than the FastSpeech2 baseline.

This paper is organized as follows: Section 2 describes our proposed model and the letter-to-sound mapping used to train our FastSpeech2 on orthographic sequences. Section 3 describes the extended dataset we used to train our model, and the training procedure. Prior to the Blizzard Challenge results, we evaluated the accuracy of the proposed phonetic prediction layer in section 3.3. Finally, results of the Blizzard evaluation are discussed in section 4.

2. Model: FastSpeech2 with mixed inputs

This section describes FastSpeech2 baseline architecture enhanced with the proposed phonetic prediction layer. The overall architecture of the proposed model is shown in Fig.1. The implementation is available online[1].

2.1. Model Architecture

The proposed model is very close to one of the open source FastSpeech2 implementation [4]. The encoder, variance adaptor and decoder are kept unchanged[2]. Following early implementations of FastSpeech2, the pitch predictor is trained on raw pitch values in semitones, instead of continuous wavelet transforms [10] in latter works. Pitch and energy values are extracted using WORLD pre-processing toolbox [11], and are averaged by phonemes, and normalized. The same multi-speaker model is used for the Hub-task and the Spoke-Task of this Blizzard Challenge. Speaker control is achieved through the addition of a trainable speaker embedding at the output of the text encoder. The model is trained on both orthographic and phonetic input sequences, following the mixed-inputs training procedure [12].

Following [9], an additional phonetic prediction layer is added at the output of the text encoder. This layer predicts a one-to-one mapping between orthographic inputs and phonetic outputs. This one-to-one letter-to-sound mapping (L2S) is further described in Section 2.2. The goals of this layer are twofold: first, it helps disambiguating homographs as shown in [13]. Second, it enables to train the text encoder

[1] https://github.com/MartinLenglet/Blizzard2023_TTS

[2] https://github.com/ming024/FastSpeech2

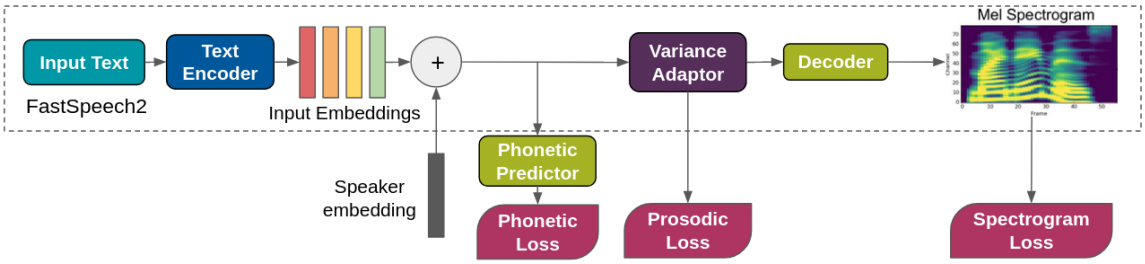


Figure 1: *Model Architecture of the multi-speaker FastSpeech2 baseline with the phonetic prediction layer. This phonetic prediction layer is trained on the output of the text encoder.*

Table 1: *Technical specificities and performances of the proposed FastSpeech2 with mixed inputs and vocoder Waveglow. Inference speed is reported as the Real-Time Factor (RTF). The loading time is the duration needed to load the model before starting the inference. This duration is not considered to compute the inference speed. Performances are computed on a single GPU Quadro RTX 8000.*

Model	# Parameters	Memory Footprint (Mbytes)	Loading Time (s)	Inference Speed (RTF)
FastSpeech2	35 630 466	1 600	4.5	1.58×10^{-2}
Waveglow	87 879 272	2 400	3	5.31×10^{-2}
Total	**123 509 738**	**4 000**	**7.5**	6.89×10^{-2}

on <orthography|phonetic> pairs without the need for corresponding audio. This eases the training of models out of audiobooks corpora, e.g. through the use of dictionaries. The cross-entropy phonetic loss trains the model on a categorization task. This loss is added to already existing MAE spectrogram-loss and MSE pitch, duration and energy-losses. The same lexicon as the Blizzard organizers was used.

Training FastSpeech2 on orthographic inputs is usually tricky, since the training of the explicit duration predictor relies on the time-segmentation of characters in the training dataset. When using phonetic sequences, every input character is attributed a unambiguous duration, either through expert analysis of the audio signal, or with automatic tools like Montreal-Forced Aligner [14]. On the other hand, in the case of opaque languages like French which require a wider visual attention span to achieve the grapheme-to-phoneme (G2P) transcription [15], it is unclear how to distribute the duration between the multiple orthographic characters involved in one phoneme, called complex phoneme in the following. Thanks to this one-to-one L2S mapping, we are able to attribute the duration to the character of interest in case of complex phonemes, and a null duration to the other characters involved. This procedure enables to train FastSpeech2 with orthographic inputs, without relying on a front-end phonetic transcription. As a result, the raw orthographic sequence is used as is during inference.

The vocoder used is Waveglow [6]. The original architecture remains unchanged[3]. The technical specificities and performances of our system are summed up in Table 1.

2.2. One-to-one Letter-to-Sound Mapping

Following the exploration of the attention map of a fully trained Tacotron2 TTS model [2], a one-to-one L2S mapping was proposed by Lenglet et al. [9]. The main results of this study are reported in this section. This mapping is deduced from the number of frames which focus on a particular grapheme in case of complex phonemes. Examples of most commonly seen patterns are given in Fig.2. Empirical rules were deduced from these observations, summed up in Table 2. The symbol /_ / is assigned

[3] https://github.com/NVIDIA/waveglow

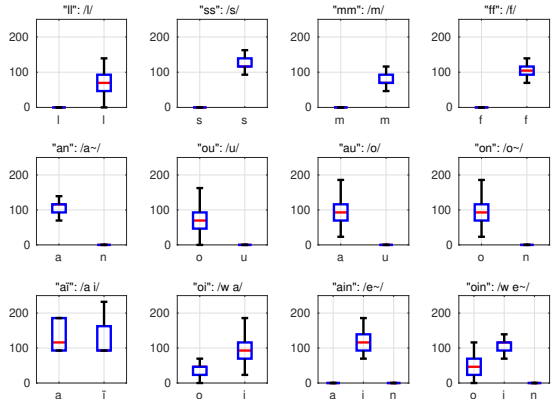


Figure 2: *Distributions of durations of activation (ms) of common character sequences in complex phonemes.*

Table 2: *Activation rules on grapheme recurrent schemes. C and V stand for consonant and vowel respectively. _ stands for muted character.*

Schemes	Activation	Examples
C C	_ C	“nn”, “ll”, “ss”
V V	V _	“an”, “ou”, “au”
V V V	_ V _	“eau”, “ain”

as output of this one-to-one mapping for muted characters.

This L2S mapping is used twice to train the model. First, characters durations when using orthographic inputs are attributed following the rules given in Table 2. This enables to train FastSpeech2 directly on orthographic sequence, and use raw orthographic sequences at inference. Thus, it enables the model to handle French liaisons on its own, which can otherwise be an issue with G2P front-end [16]. Similarly, the FastSpeech2 encoder is also able to learn how to disambiguate homographs by relying on the contextualisation provided by its successive Transformer layers [8].

Second, the phonetic prediction layer uses this L2S mapping as targets to be predicted from the input sequence in case of orthographic inputs. This phonetic prediction layer further helps the disambiguation of homographs at the output of the

encoder. In case of phonetic inputs, the input phoneme is set as the target of the prediction layer (except for punctuation marks and spaces, which are given two possible outputs: /_ / in case of null duration, or /__ / otherwise).

3. Training and Early Evaluation

3.1. Dataset

The same multi-speaker model described in section 2 was trained for both the Hub-task and the Spoke-task. To strictly follow the mixed-inputs training, utterances without phonetic alignment were excluded from the training set (19 986 out of 64 015 utterances for speaker NEB). 3 200 utterances (5% of the NEB corpus) were randomly picked in this excluded portion of the corpus as the validation set. In order to maximize the multi-speaker performances of our model, 2 additional speakers were added to the Blizzard dataset. The whole aligned corpus was included in the training set. Following Blizzard rules for the challenge, the two additional speakers are taken from open-access online databases, specified in Table 3. A part from this extended training dataset, our model is not specifically designed to achieve few-shot speaker adaptation. Nonetheless, we took part in the Spoke-Task to evaluate the benefits of our mixed representations FastSpeech2 in this context.

Moreover, since the phonetic prediction layer enables the training of the text encoder without audio recordings, we also added to the training set phonetic transcriptions from the ROBERT French-dictionary, as well as common in-context homographs. These homographs are taken from various online open-access articles, similar to [13]. The training set is further described in Table 3.

The audio output is a 80-bands Mel-spectrogram computed on the 22 050Hz audio signal with an hop-size of 256 (which is equivalent to a spectrogram sampling rate of $\approx$ 86Hz).

3.2. Training Procedure

The non-audio inputs (dictionary and homographs) are used at every stage of the training process. They are mixed with audio-inputs in each batch, with a ratio of 2/3 for audio inputs and 1/3 for non-audio inputs. While the training on non-audio inputs helps learning phonetic representations for rare words not seen in the audio corpus, this ratio minimizes the risks of degradation of the prosodic predictions and audio quality due to the absence of spectrogram-loss on the non-audio part of the corpus.

Our model was first trained following a single-speaker setup on NEB. We believe that this step helps the text encoder and decoder to focus on their primary goal which is the modulation of acoustic and prosodic local patterns according to the sequence to utter. The addition of the speaker embedding latter in the process is seen as an offset manipulation of these mean features, which is supposedly easier to learn by the model.

Table 3: *Multi-Speaker Training Dataset. Durations are given in hh:mm:ss.*

Speaker	Metadata		Audio	
	Dataset	**Gender**	**Duration**	**# Utt**
NEB	Blizzard	Female	33:33:41	44 029
AD	Blizzard	Female	2:04:53	2 515
DG	LibriVox [17]	Male	6:17:22	7 539
RO	SIWIS [18]	Female	0:35:21	586
Dictionary	Robert	-	-	95 879
Homographs	Various [13]	-	-	17 285
Total	**-**	**-**	**42:31:17**	**167 833**

The model was trained for 100 epochs on NEB only, using both orthographic and phonetic transcriptions. The batch size is set to 32. All utterances are presented twice by epoch: once with the orthographic input and once with the phonetic input. Batches are randomly selected among the whole training corpus, resulting in a mixture of speakers and input types in each batch. This mixture is not supervised.

The learning rate was fixed to 10^{-3} during this first step. Following the 2/3 - 1/3 ratio, this training includes about 50 epochs on the non-audio corpus. Following this initialization step, all other speakers were added to the training set. The learning rate exponentially decreased from this step, to reach 10^{-4} after 170 epochs. The training continued with 50 epochs on the multi-speaker corpus. When training with the multi-speaker setup, dictionary inputs are duplicated for each speaker, in order to train the phonetic predictor's dependency to the speaker. Finally, the model was trained on an evenly distributed corpus among speakers for another 50 epochs. Utterances were randomly selected to match the number of utterances in the AD corpus, when enough utterances were available. All utterances from RO were kept for this final training step. We empirically found that this final step helps modelling rarest speakers behaviors instead of copying the behavior of the most seen speaker.

The vocoder Waveglow [6] was fine-tuned from the pre-trained model shared with the GitHub implementation. The fine-tuning was performed on the NEB corpus, first for 50 epochs on the Ground-Truth spectrograms, and then for 50 additional epochs on spectrograms predicted by the FastSpeech2 model.

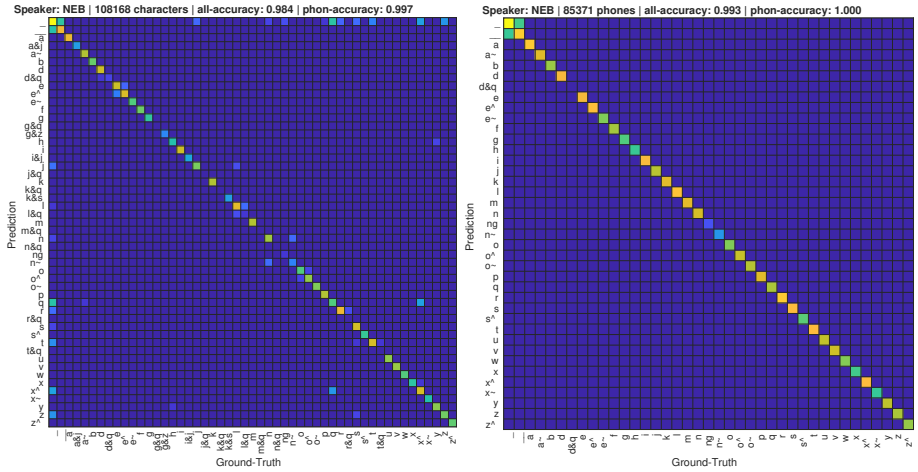
3.3. Phonetic Prediction Evaluation

On top of the evaluation performed for the Blizzard Challenge, we evaluated the performances of the phonetic prediction layer, as an indicator of the potential benefits of the proposed architecture compared to the traditional FastSpeech2 training.

As a test set, we randomly extracted 2230 additional utterances recorded by the same NEB speaker from the original M-AILABS corpus [19]. These utterances are not part of the dataset shared by Blizzard organizers, thus they have not been seen by the model during the training phase.

The phonetic prediction was computed on this test set, and confusion matrices are reported in Fig.3, using orthographic inputs (Fig.3a) and phonetic inputs (Fig.3b). Among the 108168 orthographic characters of this test set, the overall accuracy reaches 0.984 (0.997 when excluding muted characters). Interestingly, most remaining errors are confusions between close phonetic variants: mid-closed vowels VS mid-opened vowels, and full closed vowels VS semi-vowels. Most errors with muted characters are miss-predicted liaisons on ending /r/, /t/ or /z/. Note that the errors highlighted here may just reflect divergences between the Ground-Truth and the model decision on optional liaisons. On the other hand, when using phonetic inputs, this prediction is almost flawless, reaching 0.993 overall, and 1.00 when excluding spaces and punctuation marks.

While this evaluation of the phonetic accuracy of the proposed model is promising regarding the production of heterophonic homographs and the intelligibility of semantically unpredictable sentences, these tasks are specifically designed to test the model out of what has been seen during the training. Thus, results may differ on these specific tasks.



(a) *Orthographic Inputs* (b) *Phonetic Inputs*

Figure 3: *Confusion Matrices of the phonetic prediction layer, for orthographic inputs (left) and phonetic inputs (right).*

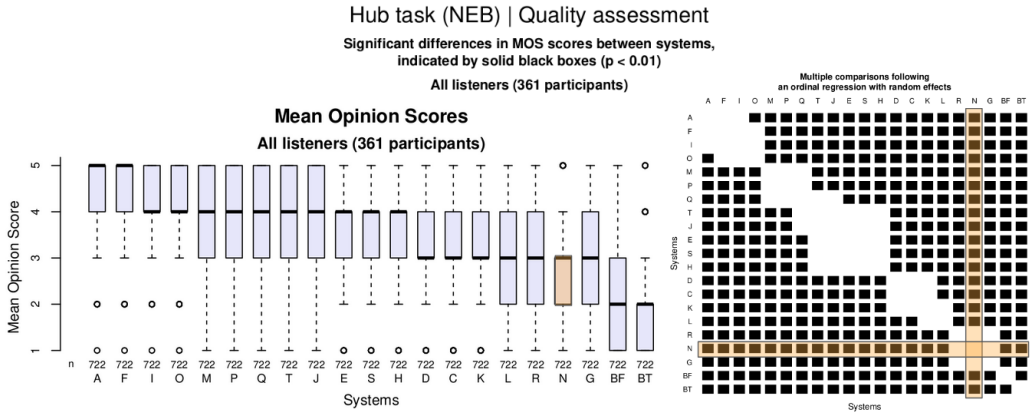


Figure 4: *Mean Opinion Scores (MOS) for the Hub Task. Our model N is highlighted in orange. The left graph shows MOS by system. In the right graph, black squares show that the difference between the two models is significant (p<0.01).*

4. Blizzard Results

This year Blizzard Challenge evaluates speech produced by TTS on multiple criteria. The Hub-Task evaluates models capacities to reproduce natural behaviors of NEB. The naturalness is evaluated with Mean Opinion Scores (MOS). Intelligibility is evaluated on heterophonic homographs disambiguation and Semantically Unpredictable Sentences (SUS). Similarly, the Spoke-task evaluates the ability of the model to produce natural voice with few examples on AD. Our system did not perform better than the FastSpeech2 baseline in this speaker adaptation task. Thus, this section is focused on the most interesting results of our proposed system: naturalness and intelligibility on the Hub-task. Results commented in this section have been computed regardless of listeners experience in the domain. Among all presented systems, A is the original recording, BT is the baseline Tacotron2, BF is the baseline FastSpeech2, and N is our proposed model. Our model N is highlighted in orange in all figures.

4.1. Naturalness on the most seen speaker

The results of the naturalness assessment on the Hub-Task are reported in Fig.4. Although not showing impressive results, our model was significantly preferred over the FastSpeech2 baseline. Training our model on orthographic sequences may have helped to produce more accurate phonetic patterns. In comparison, the FastSpeech2 Baseline BF has been trained solely on phonetic inputs. Thus, BF relies on a G2P front-end to convert the orthographic sequences of the test set before synthesis. Depending on the front-end used, it may produce errors, in particular with French liaisons which may be hard to predict.

We also believe that our overall MOS score could have benefited from simple post-treatments to reduce the produced noise. We are aware that our Waveglow vocoder produces background noise which can be detrimental to listeners judgment. However, in an attempt to avoid the use of heuristics, we decided to enter the challenge without post-processing denoising techniques.

4.2. Heterophonic Homographs Disambiguation

Intelligibility assessment on heterophonic homographs is reported in Fig.5. Our model N achieves an average score among all systems. Our model shows global improvements over the BF, which was expected thanks to the addition of mixed representations and the training of the phonetic prediction layer on the auxiliary dictionary and homographs corpus.

More specifically, our model performs very well on homographs that have been seen with enough examples in its homograph corpus. “Fils” (261 examples) pronounced /f i s/: “son (en)” VS /f i l/: plural of “fil (fr)”, “wire (en)” has a intelligibility score or 100% for both variants, whereas systems with over-

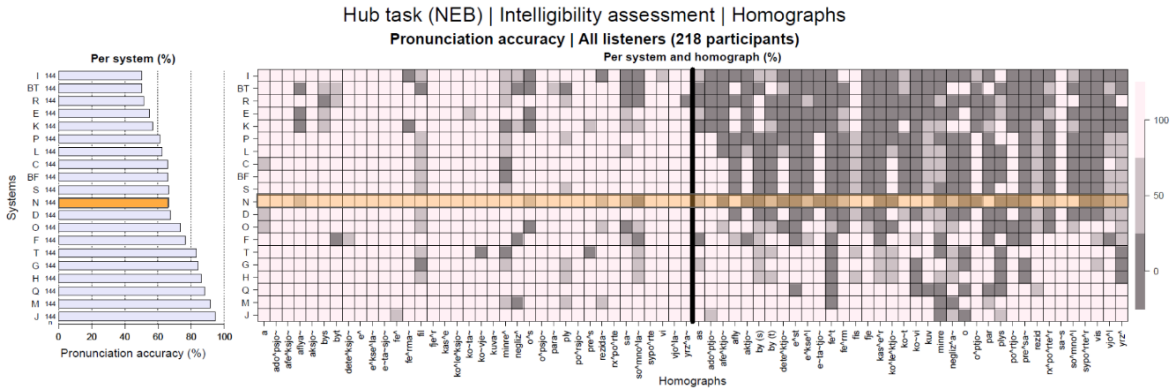


Figure 5: *Homographs intelligibility scores for the Hub Task. Our model N is highlighted in orange. The right graph shows the percentage of correct pronunciation by system. The right graph shows this intelligibility assessment by homograph.*

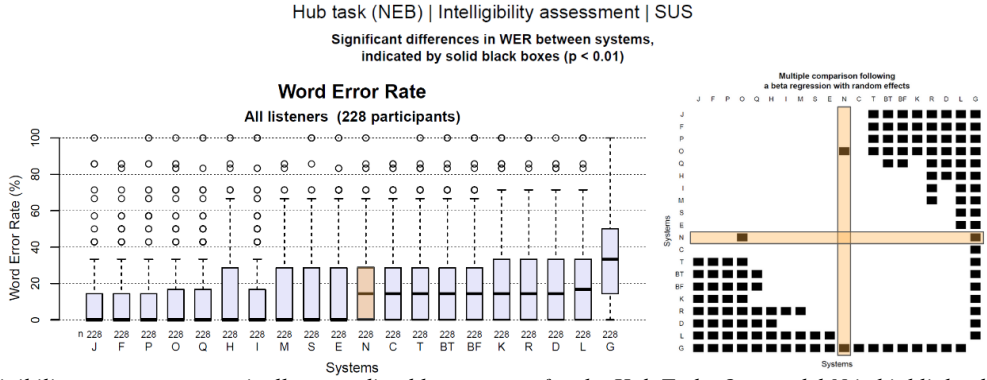


Figure 6: *Intelligibility scores on semantically unpredictable sentences for the Hub Task. Our model N is highlighted in orange. The right graph shows the percentage of correct pronunciations by system. In the right graph, black squares show that the difference between the two models is significant (p<0.01).*

all better scores do not achieve such accuracy on this specific homograph. This is also true for "convient" (181 examples) or "fier" (366 examples) (/k o~v i e~/: "suit (en)" VS /k o~v i/: "invite (en)" — /f j eˆr/: "proud (en)" VS /f j e/: "trust (en)"), with the most common forms /k o~v i e~/ and /f j eˆr/ being systemically pronounced by other TTS regardless of the context. On the contrary, "options" (117 examples), "intentions" (141 examples) and "portions" (145 examples) also appear in the homographs training corpus, but with fewer examples. The number of examples and the balance between variants impact the performances of the system. However, the proposed method helps modelling homographs if enough examples are given during training.

4.3. Semantically Unpredictable Sentences (SUS)

Intelligibility scores on SUS are reported in Fig.6 for all systems. All models but one perform similarly on SUS. Our system only statistically differs from G which shows the worst results on this task, and from O which performs better. On the other hand, BF is found to statistically differ from the top 5 performing systems. The mixed representations and phonetic prediction layer may have help to achieve this task.

5. Conclusions and Discussion

This paper has described the GIPSA-Lab system for the Blizzard Challenge 2023. This system is very similar to the original FastSpeech2 architecture, with two major additions: the training on orthographic sequences and the phonetic prediction layer. The phonetic prediction layer was evaluated before the Blizzard Challenge, and showed very promising performances. The results of the proposed system in Blizzard evaluation confirm the benefits of these additions compared to the baseline FastSpeech2 system. Our system performed better than the baseline FastSpeech2 on naturalness and intelligibility on the most seen speaker in the corpus. On the other hand, our system did not show much difference in terms of speaker adaptation.

The results of the disambiguation of heterophonic homographs shows the potential of the proposed training of the text encoder on <orthography|phonetic> pairs without the need for audio recordings. However, disambiguation was only improved for the most seen examples in the homographs training corpus. Wider corpora may help to achieve better results. The training procedure may also impact the final result. The ratio of non-audio inputs in the training batches may vary to include more phonetic training during the learning phase.

The vocoder used also contributed to the mitigated MOS evaluated during quality assessments. We experience mitigated audio quality with Waveglow, which tends to add background noise in our samples. The impact of this noise can be reduced with post-processing denoising, that we did not explore in our Blizzard submission. Other vocoders like Hifi-GAN may also help regarding this issue.

6. Acknowledgements

This research has received funding from the BPI project THERADIA and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). This work was granted access to HPC/IDRIS under the allocation 2023-AD011011542R2 made by GENCI.

7. References

[1] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[3] T. Kenter, V. Wan, C.-A. Chan, R. Clark, and J. Vit, "Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3331–3340.

[4] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*, 2021.

[5] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[6] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*. IEEE, 2019, pp. 3617–3621.

[7] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[9] M. Lenglet, O. Perrotin, and G. Bailly, "Modélisation de la parole avec tacotron2 : Analyse acoustique et phonétique des plongements de caractère," in *Actes des Journées d'Etudes sur la Parole (JEP)*, Noirmoutiers, France, June 13-17 2022.

[10] M. Vainio, A. Suni, and D. Aalto, "Continuous wavelet transform for analysis of speech prosody," *Tools and Resources for the Analysys of Speech Prosody (TRASP)*, 2013.

[11] M. Morise, H. Kawahara, and H. Katayose, "Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, 2009.

[12] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.

[13] M.-L. Hajj, M. Lenglet, O. Perrotin, and G. Bailly, "Comparing nlp solutions for the disambiguation of french heterophonic homographs for end-to-end tts systems," in *SPECOM*. Springer, 2022, pp. 265–278.

[14] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[15] A. W. Black, K. Lenzo, and V. Pagel, "Issues in building general letter to sound rules," in *The third ESCA/COCOSDA workshop (ETRW) on speech synthesis*, Jenolan Caves House, Blue Mountains, Australia, 1998.

[16] G. Bailly, M. Lenglet, O. Perrotin, and E. Klabbers, "Advocating for text input in multi-speaker text-to-speech systems," in *12th ISCA Speech Synthesis Workshop*. ISCA, 2023, pp. 13–18.

[17] J. Kearns, "Librivox: Free public domain audiobooks," *Reference Reviews*, 2014.

[18] P.-E. Honnet, A. Lazaridis, P. N. Garner, and J. Yamagishi, "The siwis french speech synthesis database. design and recording of a high quality french database for speech synthesis," Idiap, Tech. Rep., 2017.

[19] I. Solak, "The M-AILABS speech dataset," https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/, 2019.

12th ISCA Speech Synthesis Workshop (SSW2023)
26-28 August 2023, Grenoble, France

Local Style Tokens: Fine-Grained Prosodic Representations For TTS Expressive Control

Martin Lenglet, Olivier Perrotin, Gérard Bailly

Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, France
{martin.lenglet,olivier.perrotin,gerard.bailly}@grenoble-inp.fr

Abstract

Neural Text-To-Speech (TTS) models achieve great performances regarding naturalness, but modeling expressivity remains an ongoing challenge. Some success was found through implicit approaches like Global Style Tokens (GST), but these methods model speech style at utterance-level. In this paper, we propose to add an auxiliary module called Local Style Tokens (LST) in the encoder-decoder pipeline to model local variations in prosody. This module can implement various scales of representations; we chose Word-level and Phoneme-level prosodic representations to assess the capabilities of the proposed module to better model sub-utterance style variations. Objective evaluation of the synthetic speech shows that LST modules better capture prosodic variations on 12 common styles compared to a GST baseline. These results were validated by participants during listening tests.

Index Terms: speech synthesis, expressive TTS, style control, prosody modeling

1. Introduction

Latest neural Text-to-speech models (TTS) [1, 2], combined with neural vocoders [3, 4] achieve high standards in terms of naturalness. However, these systems still struggle to model the variability of expressive speech. Two main factors are pointed out to explain these difficulties: 1) the lack of labelled data and 2) the design choice of architecture which enables to learn this variability, as well as its control at inference time. One of the most successful model to tackle both issues is the Global Style Token (GST) architecture [5]. The GST design relies on an reference encoder [6], which converts a reference audio sample into a fixed-size vector which summarizes paralinguistic information. A set of unconstrained tokens are simultaneously trained as an attempt to disentangle main speech features within this paralinguistic representation. Although this architecture enables training on data that is not expressive-labeled, unconstrained tokens are hard to interpret, and post-hoc analysis is necessary to efficiently control the desired synthesis style [7].

Later studies elaborated on improving the expressive control provided by such architectures. Through supervised training [8] or automatic exploration of latent spaces [9], these progress have enabled the careful design of the utterance-wise style bias to be applied in order to generate speech following the target style, without the need for an explicit audio reference. However, natural expressive speech relies on multiple levels of variations. The prosodic structure of one's speech not only depends on one's intents or style, but also on the content itself, as syntactic and semantic structures play an important role in the organization of stress and phrasing [10, 11]. As a result, utterance-wise style embeddings may lack finer-grained representations in order to fully mimic natural voice behavior.

In this paper, we propose to model fine-grained prosodic patterns through an auxiliary module called ***Local Style Tokens (LST)***. Extending the GST implementation on a segmental level, this module learns to model the residual local speech variations that remain to be explained after utterance-style bias is applied. The proposed module can be applied at multiple scales, providing that such scale can be automatically inferred from the textual input. In this paper, this module was evaluated on Word-level and Phone-level. After discussing related works in Section 2, Section 3 describes the LST specificities and implementation. Objective evaluations described in Section 4.2 compare this module's performances with the GST utterance-wise control and the natural speech. Finally, Section 4.3 describes the listening test procedure we conducted and its results.

2. Related Work

Fine-grained prosodic representations have been proposed in TTS before. By construction, pitch and energy embeddings in FastSpeech2 variance adaptor [2] are spectrogram frame-level prosodic embeddings. These provide some prosodic control at inference, but also helps better modeling fundamental frequency. The LST module relies on the same mechanism as prosodic predictors, by re-injecting prosodic representations within the model in the layer they are predicted from.

More focused toward expressive control, [12] proposed to enhance Tacotron2 [1] with word-level style embeddings that are concatenated to the encoder output. Word-level representations are computed with recurrent layers, and then passed to a style attention layer similar to GST [5]. This work inspired us for the present study, but we tried to avoid its main limitation: authors had to train a Prior Encoder, which predicts word style embeddings from the text input in order to synthesize text without audio reference. As a result, the output synthesis is solely based on the text input, denying the choice of expressive style at inference. On the contrary, we aim to use Word (or Phone)-level information to locally refine an global utterance-wise style bias, and therefore combine both style and content inputs.

Hierarchical TTS models like CHiVE [13] or MsEmoTTS [14] also take advantage of the multi-level aspect of speech, by combining intermediate representations from different scales: phonemes, syllables, words, utterance, etc. The entire architectures of these models are built on this hierarchical representations. On the other hand, the proposed LST module is independent; it can be plugged to any encoder-decoder TTS architecture, with various scopes of representation.

10.21437/SSW.2023-19

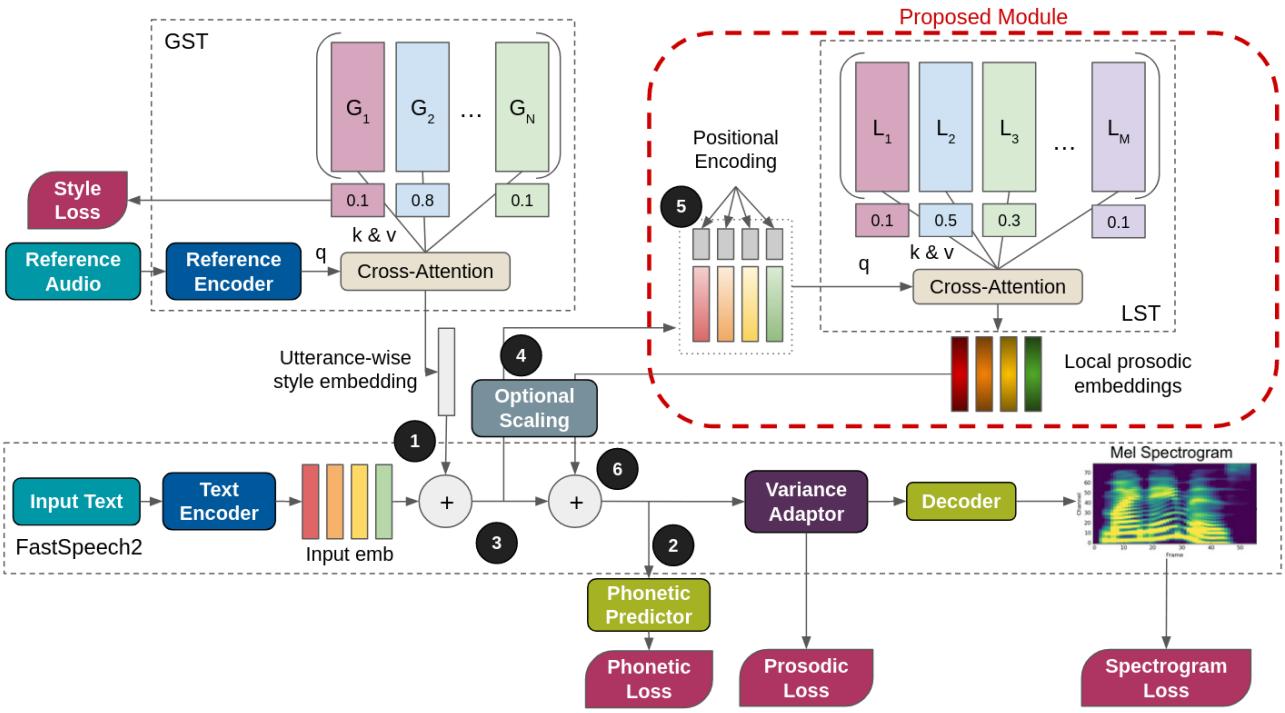


Figure 1: *Model Architecture. The Local Style Tokens module (LST) is plugged after the addition of the utterance-wise style bias.*

3. Proposed Model

This section describes the architecture of the proposed Local Style Tokens (LST) module and how it is integrated in the GST-enhanced FastSpeech2 pipeline. The overall architecture of the proposed model is shown in Fig.1.

3.1. Model Architecture

3.1.1. Model Backbone

The backbone of the model is FastSpeech2 [2], whose encoder, variance adaptor and decoder are kept unchanged[1]. In addition, a label-constrained GST module [8] is plugged at the output of the text encoder (Fig 1.1)[2]. This constrained-GST module converts a reference audio sample into a fixed-size vector through a reference encoder [6]. This fixed-size vector is then used as the query of the cross-attention mechanism in an emotion token layer. Similarly to GST [5], this emotion token layer computes weights that measure the similarity between the reference vector and each global style token. Following [8], a cross-entropy loss is added to enforce each token to encode one particular style. The weighted sum of tokens is then added to all phoneme embeddings computed by the text encoder. Contrary to a set of learnable style embeddings, this module helps training the model on heterogeneous style samples. When given the same style as target, one speaker may produce highly variable utterances, with varying intensity of the given style. The constrained-GST module may account for the intensity by using a mixture of tokens for low intensity utterances, even though their label is the same as unambiguous utterances.

Following [15], a phonetic prediction layer is also added at the output of the text encoder (Fig 1.2). This layer predicts a one-to-one mapping between orthographic inputs and phonetic outputs. The goals of this layer are twofold: first, it helps disambiguating homographs as shown in [16]. Second, it enables to train the text encoder on <orthography|phonetic> pairs without the need for corresponding audio. This eases the training of models out of audiobooks corpora, e.g. through the use of dictionaries.

3.1.2. Local Style Token Module

The Local Style Tokens architecture (LST) is introduced as an auxiliary module which further modulates the output of the text encoder. Although this module does not need to be combined with the GST module, the LST layer alone does not provide explicit control of the synthesis style at inference, which is why the constrained-GST is used in this model.

In FastSpeech2 [2], the variance adaptor implements three prosodic predictors which predict duration, pitch and energy from the output of the text encoder. The prosodic losses associated to these predictors constrain the latent space to encode at least representations of these three prosodic features. Similarly, style embeddings [5, 17] that are added to all phoneme embeddings suppose that additional acoustic and prosodic features are at least partially encoded in this latent space. The LST module may be seen as a residual layer which modulates the latent representations that have been uniformly biased by the GST embedding, according to the content or the position of linguistic units in the utterance. This modulation further improves acoustic and prosodic representations in this latent space.

The LST layer follows the same architecture as the original GST [5]. Two levels of local tokens are examined in this study: Word-level and Phone-Level. In the case of Phone-level tokens, this module takes as inputs the globally biased phoneme embeddings sequence (Fig 1.3). For Word-level, this sequence is averaged by word, to compute word-level representations (Fig 1.4). Because our dataset preserves word boundaries and punctuation marks in case of phonetic inputs, pseudo-word representations

[1] https://github.com/ming024/FastSpeech2

[2] GST implementation based on https://github.com/taneliang/gst-tacotron2

are also computed for spaces and punctuation marks (or both when consecutive), also by averaging embeddings.

This input is enhanced by a 32-dimensional positional embedding [18], which is concatenated (Fig 1.5). Indeed, similarly to GST [5], the cross-attention mechanism in the LST layer uses dot product attention, which cannot infer relative positions of representations in the input sequence, in opposition with recurrent networks. However, acoustic patterns relative to style generation depends on the syntactic structure of the utterance and on the relative position of units in the utterance. Although such positional encoding has already been added to phoneme embeddings in the text encoder, preliminary studies showed benefits of explicitly enhancing representations with positional encoding.

This input tensor serves as a set of queries for the cross-attention mechanism in the LST layer. A set of weights is computed for each element in the sequence, and the weighted sum of token values constitutes the local prosodic embeddings sequence which is added to the globally biased phoneme embeddings before the variance adaptor (Fig 1.6). In case of Word-level LST, the local prosodic embedding is first duplicated to be added to all phonemes in the given word (resp. pseudo-word). For ease of interpretability of local token weights, the cross-attention mechanism is single-headed.

3.2. Training and Inference Processes

During training, the reference mel-spectrogram matches the target output. The reference encoder and the cross-attention GST work as an emotion recognition module which computes a probability distribution of the given audio input on all constrained style tokens. In contrast, the LST weights are not constrained during training. The LST layer does not require additional loss. It is trained by the back-propagation of the spectrogram loss, prosodic predictors losses and phonetic loss. The back-propagation is not stopped at the input of the LST module, which enables the text encoder to incorporate features that may be used to compute local prosodic embeddings in the LST layer. The entire model can be trained simultaneously, from scratch.

Similarly to constrained-GST, two style control methods are available at inference: 1) use a target reference audio which produces a mixture of global style tokens or 2) specify the mixture of global style tokens to use. Because the GST module is constrained, each global style token has been trained to produce one particular style. Thus, one-hot vectors are particularly fitted to generate the desired style. Local prosodic embeddings are computed in parallel by the LST module, which does not impact the inference speed of the model.

4. Experiments and Results

4.1. Models and dataset

Three models are trained for this study: 1) FastSpeech2 with constrained-GST referred as ***GST*** (the Baseline) ; 2) Baseline enhanced with word-level LST referred as $\boldsymbol{LST_W}$; and 3) Baseline enhanced with phoneme-level LST referred as $\boldsymbol{LST_P}$. $\boldsymbol{LST_W}$ offers more context at the input of the LST module, which may result in a more careful choice of representations in the LST layer. On the other hand, word style bias may result in less intra-word modulation.

All models are trained on the same dataset, given in Table 1. This internal French dataset has been uttered by a French professional theater actress. Sentences are taken from the SIWIS database [19], which is composed of isolated extracts from French Novels and French parliament debates. For expressive speech recording, she was asked to utter the given sentences with the specified style during exercise-in-style sessions. During these sessions, the actress was prompted to start her utterance with a context sentence relative to the style being produced: "I am begging you" for "Pleading", "I do not believe it" for "Skeptical", "Really?" for "Surprised", etc. This context sentence was cut from the final recording. The recordings are being evaluated to verify that the produced style is correctly recognized by naive speakers, but this evaluation is still on-going at the time of writing of this study.

The content is decorrelated from the expressed style, and sentences differ between styles. Sentences that were not uttered with a specific style were labeled as "Narrative". This audio-visual expressive dataset was recorded in the GIPSA-Lab, as part of the Theradia project [20]. The 12 styles were chosen to cover the expressive needs of the Theradia application. Only the audio was used in this study. 5% of the corpus was randomly selected as the test set. All models are trained for 250 epochs using both orthographic and phonetic input representations. Following early implementations of FastSpeech2, the pitch predictor is trained on raw pitch values in semitones, instead of continuous wavelet transforms [21] in latter works. Pitch and energy values are averaged by phonemes, and normalized. The one-to-one phonetic targets for the phonetic prediction task are established using patterns described in [15]. The vocoder used is Waveglow [3].

Table 1: *Expressive Dataset. Durations are given in minutes.*

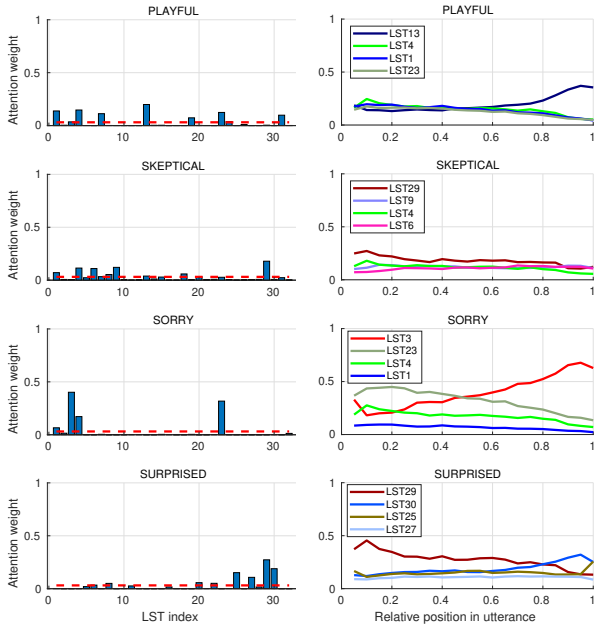
Style	**Train**		**Test**	
	Duration	**# Utt**	**Duration**	**# Utt**
Angry	24.2	523	1.5	32
Comforting	32.3	488	1.6	27
Committed	21.1	430	1.4	29
Enthusiastic	29.5	569	1.4	28
Obvious	27.0	492	1.5	27
Playful	19.1	465	1.5	28
Pleading	34.2	605	1.9	31
Skeptical	29.8	620	1.6	32
Sorry	24.2	448	1.1	23
Surprised	26.8	503	1.6	32
Thoughtful	43.4	450	2.1	27
Narrative	287.6	6235	14.6	307
Total	**599.2**	**11828**	**31.8**	**633**

Table 2: *Number of Local Style Tokens used by the model per style.*

Style	$\boldsymbol{LST_W}$		$\boldsymbol{LST_P}$	
	# Tokens	**# Exclusive**	**# Tokens**	**# Exclusive**
Angry	9	1	8	0
Comforting	8	1	7	0
Committed	8	0	11	0
Enthusiastic	6	0	10	0
Obvious	7	0	8	0
Playful	9	1	11	0
Pleading	6	0	10	0
Skeptical	10	0	12	0
Sorry	4	0	8	0
Surprised	8	0	11	0
Thoughtful	8	0	12	0
Narrative	11	3	13	2
Overall (/32)	**32**	**6**	**30**	**2**

Table 3: *Mean errors per style computed on the test set. Blue (resp. red) indicates a lower error (resp. higher error) than* ***GST****. * and ** indicate that the distribution statistically differs from the* ***GST*** *baseline with p<0.05 and p<0.01, respectively.*

Style	**Spectral Error (dB)**			**Duration Error (ms)**			**Pitch Error (Semitones)**			**Energy Error (dB)**		
	GST	***LST$_W$***	***LST$_P$***	***GST***	***LST$_W$***	***LST$_P$***	***GST***	***LST$_W$***	***LST$_P$***	***GST***	***LST$_W$***	***LST$_P$***
Angry	0.93	0.91	0.93	9.18	9.01	9.13	2.29	2.14	2.50	3.24	3.15	3.34
Comforting	0.88	0.88	0.89	11.41	11.64	11.12	1.59	1.62	1.77	3.12	3.24	3.11
Committed	1.00	0.99	0.96	9.83	9.83	9.33	4.10	4.15	3.93	3.26	3.16	3.24
Enthusiastic	1.18	1.16	1.18	9.82	9.45	9.82	4.31	3.93	4.31	3.03	3.10	3.06
Obvious	1.07	1.05	1.08	10.74	10.23	10.01	3.32	3.12	3.41	3.12	3.09	3.12
Playful	0.97	0.99	0.96	12.08	11.29	11.25	4.06	3.98	4.11	3.09	3.06	3.04
Pleading	0.92	0.92	0.91	9.67	9.03	9.49	1.93	1.78	1.72	2.49	2.44	2.45
Skeptical	0.98	0.97	0.99	10.10	9.85	10.13	2.86	3.03	3.08	3.06	3.03	**3.23****
Sorry	0.67	0.68	0.67	9.50	9.50	9.70	1.05	1.16	1.04	2.64	2.75	2.77
Surprised	0.97	0.97	0.97	10.20	9.85	10.07	3.47	3.33	3.40	3.21	3.13	3.30
Thoughtful	0.94	0.95	0.97	22.23	22.28	22.52	2.51	2.63	2.52	2.86	2.91	2.96
Narrative	0.90	0.90	0.89	10.45	**10.36***	10.53	2.75	2.73	2.74	2.93	**2.86***	**2.86****
Total	0.93	0.93	0.92	10.52	10.31	10.44	2.70	2.67	2.71	2.97	2.94	2.96



(a) *Mean LST usage by style.* (b) *Mean LST usage relative to the position in utterance (0: first character, 1: last character).*

Figure 2: *Local Style Tokens usage by style for* ***LST$_W$****. Four styles are shown as examples: "Playful", "Skeptical", "Sorry" and "Surprised". Only the 4 local tokens with the maximum mean attention weights are shown in Fig 2b.*

Following the constrained-GST architecture given by [8], 12 tokens are needed in the ***GST*** layer to account for each style label cited above. The target styles given to the actress are used as style labels. The number of local style tokens is fixed to 32 for both ***LST$_W$*** and ***LST$_P$***. 32 tokens is chosen as a middle ground between sharing tokens across GST representations and providing enough local tokens so that each global style can rely on dedicated local tokens. To evaluate the usage of each individual local token, 100 utterances of the test set were randomly selected, and each utterance was generated with the 12 styles of the corpus. Mean attention weights of each local token were computed per style. Examples of mean attention weights by local token and the dynamic of such attention are given in Fig 2 for ***LST$_W$***. Table 2 summarizes the number of local tokens used per style, as well as the number of tokens that are exclusive to the specified style. One local token is counted as used if its mean activation weight is above the uniform distribution across all local tokens (above the red dashed line in Fig 2a). The overall number of tokens used differs from the sum because some tokens are shared across styles. Two tokens are never used by ***LST$_P$***. ***LST$_W$*** and ***LST$_P$*** with 64 tokens were tested but showed that too many tokens were never used.

The diversity of local tokens usage illustrates the benefits of modelling prosody at a smaller scale. Multiple local tokens are used by all styles to model various local patterns. "Angry", "Comforting", "Playful" and "Narrative" use exclusive local tokens in ***LST$_W$***, assessing for unique speech behaviors in this sub-corpus (same for "Narrative" in ***LST$_P$***). Figure 2b shows the dynamic of local tokens attention relative to the position in the utterance. Global styles exhibit various patterns, but most characteristic behaviors are found at the beginning (LST29 for "Surprised") and at the end of utterances (LST13 for "Playful" and LST3 for "Sorry"). Other styles like "Skeptical" are more stable, but smaller variations of local tokens usage also indicate that the LST module helps modulating representations at a finer-grain.

4.2. Objective Evaluation

Objective evaluations of the synthetic models were conducted to assess the benefits of the proposed model compared to the baseline. Models are evaluated on 3 aspects: training loss criteria, pitch variations and phrasing behaviors. All statistical differences between distributions are evaluated pair-wise through non-parametric Wilcoxon rank sum tests. The objective metrics shown in this section focus on various evaluations of the three main prosodic features: duration, pitch and energy. Other acoustic features like voice quality may impact style modelling [22], but were not measured in this study.

4.2.1. Test Set Errors

All models are trained under the same loss criteria, which include mel-spectrogram losses and prosodic features predictions (duration, pitch and energy). Table 3 summarizes these errors from the test set ground truth (***GT***) after training. Spectral error

Table 4: *Mean standard deviation of pitch per style (Semitones). * indicates that the distribution statistically differs from the **GT** ($p<0.05$). Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the **GST** baseline.*

Style	***GT***	***GST***	***LST$_W$***	***LST$_P$***
Angry	3.59	**2.95**	**2.85***	**2.77***
Comforting	1.94	1.68	1.70	1.77
Committed	3.92	3.92	3.94	3.69
Enthusiastic	4.79	**3.27***	**3.44***	**3.24***
Obvious	4.25	**3.06***	**3.41***	**3.39***
Playful	5.27	**4.15***	**4.03***	4.04*
Pleading	2.90	2.36	2.45	2.51
Skeptical	4.49	**2.90***	**3.36***	**3.04***
Sorry	1.85	**1.49***	**1.65**	**1.58***
Surprised	5.66	**4.03***	**4.20***	**4.08***
Thoughtful	2.70	2.53	2.64	2.54
Narrative	4.94	**3.89***	**3.91***	**3.89***

is computed on synthesis aligned with Dynamic Time Warping (DTW) [23]. Mean euclidean distances are evaluated on the alignment path. Duration and energy errors are computed on all phonemes, while pitch error is only evaluated on vowels.

Lower errors indicate that models that implement the LST modules produce speech closer to the ***GT*** for most styles. Over all errors, ***LST$_W$*** provides the most consistent benefits, with 28 improvements and 14 degradations, compared to 20 improvements and 20 degradations for ***LST$_P$***. These improvements were significant for "Narrative", but not for the other styles. "Committed", "Enthusiastic", "Pleading", "Surprised" and "Narrative" are the most improved styles. This indicates that those five styles rely on local prosodic patterns that are difficult to model with utterance-wise style representation. On the other hand, "Comforting", "Skeptical", "Sorry" and "Thoughtful" show higher errors with LST. Overall, the more mitigated results of ***LST$_P$*** may be explained by the wider variability provided by local tokens at the phoneme scale. This variability opens the door for more risks of divergence with ***GT***.

While lower errors indicate that synthetic speech is closer to the natural utterances recorded in our corpus, there is no golden standard for conveying a given style. Many variants: 1) could have been performed by the recorded speaker for this same sentence and style, and 2) may be perceived as similarly expressive for a human listener. As a result, the ***GT*** is not the only licit speech production, and more objective evaluations are needed to assess the expressive quality of the synthetic speech. In the following, we then compare distributions of prosodic parameters measured on ***GT*** and on each of our models. Our criteria for a successful rendering of prosodic features is therefore to have **non**-significant differences between a model and the ***GT***.

4.2.2. Pitch standard deviation

Pitch standard variations by utterance is commonly used to evaluated expressive capabilities of TTS models [2, 12]. Table 4 compares the pitch variability of ***GT*** to that of the synthetic models. Highly variable styles like "Enthusiastic", "Obvious", "Playful", "Skeptical" and "Surprised" are harder to model for TTS, as shown by statistical differences between ***GT*** and all synthetic models. Overall, the LST module helps generating more pitch variability, even though results were not significant. Significant improvements were found for "Sorry", with ***LST$_W$*** generating pitch standard deviations closer to ***GT***.

Table 5: *Mean proportion of silences in synthetic vs. **GT** utterances (in %). ** indicates that distributions statistically differ from the **GT** with $p<0.01$. Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the **GST** baseline.*

Style	***GT***	***GST***	***LST$_W$***	***LST$_P$***
Angry	2.6	**2.0****	**1.8****	**3.0**
Comforting	2.5	2.3	2.2	1.9*
Committed	4.8	3.3	3.4	3.3
Enthusiastic	1.6	1.7	1.6	1.2
Obvious	0.8	1.1	0.8	1.0
Playful	6.6	4.7	4.8	5.2
Pleading	1.3	**0.6****	**0.6****	**0.7****
Skeptical	2.4	**1.8****	**2.0****	**1.5****
Sorry	2.2	**1.4****	**3.2****	**1.9****
Surprised	2.0	1.9	1.4	1.0
Thoughtful	1.8	2.4	1.4	1.6
Narrative	3.8	**2.5****	**2.6****	**2.6****

4.2.3. Phrasing Error

Phrasing is decisive in perceptual judgements [24, 25]. Notably, varying frequency of silences when modifying the speaking rate is a key feature of natural voice that synthetic models generally struggle to achieve. Table 5 shows mean silence proportions per style for each model and ***GT***. Significant differences between ***GT*** and synthetic models for "Pleading", "Skeptical", "Sorry", and "Narrative" demonstrate the difficulties of TTS to replicate natural balance between speech and silences for these styles. The LST module does not provide much improvement in that regard. Conversely, ***LST$_P$*** produces more pauses for styles with high silences ratio like "Angry" and "Playful", whose natural behaviors are hardly replicated by utterance-wise style bias in ***GST*** (this improvement was significant for "Angry").

Duration modulation were also evaluated as an indicator of local prosodic patterns. We hypothesize that polysyllabic words should be more impacted by local modulations, as they are mostly content words. At least some of the studied styles should emphasize local key points in the utterances that are embodied by content words. Word duration modulation is evaluated as the ratio between the duration of the last vowel and the mean duration of other vowels of the same word. This measure indicates the lengthening of last syllable of polysyllabic words, as approximation of content words. Table 6 summarizes evaluated duration modulation per style. Lengthening of the last syllable of polysyllabic words is very common in ***GT***, as shown by mean word duration modulations above 1.25 for every style. "Obvious", "Pleading", "Sorry" and "Thoughtful" show the higher degree of modulation. This modulation is closely replicated by all synthetic models, with slight variations between models. Interestingly, ***GST*** tends to elongate durations excessively, in particular on "Enthusiastic", "Playful" and "Thoughtful", while the LST modules help producing more natural duration modulations.

Table 6: *End syllable duration modulation evaluated on polysyllabic words. * indicates that the distribution statistically differs from the **GT** ($p<0.05$). Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the **GST** baseline.*

Style	***GT***	***GST***	***LST$_W$***	***LST$_P$***
Angry	1.34	1.34	1.28	1.34
Comforting	1.33	1.35	1.39	1.36
Committed	1.34	1.34	1.39	1.34
Enthusiastic	1.36	1.47	1.44	1.38
Obvious	1.41	1.43	1.41	1.40
Playful	1.32	1.54	1.44	1.51
Pleading	1.37	1.30	1.35	1.30
Skeptical	1.24	1.28	1.31	1.22
Sorry	1.43	1.44	1.52	1.46
Surprised	1.25	1.23	1.27	1.21
Thoughtful	1.88	1.95	1.91	1.95
Narrative	1.33	**1.36***	**1.36***	**1.37***

4.3. Listening Experiment

In order to evaluate perceptual differences between the proposed model and the baseline, 60 participants took part in an online MUSHRA-like experiment [26], run with the framework webMUSHRA [27]. Given the text uttered and the target style, participants were asked to evaluate on a scale from 0 (very bad) to 100 (excellent) if the style was correctly rendered. For this listening test, we selected 10 utterances per style that maximize spectral distances between systems (120 in total). 5 groups of 12 participants evaluated each 24 utterances (2 per style), with 5 systems per utterance: the ***GST***-enhanced FastSpeech2 baseline, the two proposed models ***LST$_W$*** and ***LST$_P$***, the vocoded ***GT*** (high anchor), and a FastSpeech2 without GST trained on non-expressive data (low anchor) referred as ***LA***. Because the Ground-Truth is not the only way to convey the given style, it was not given as an explicit reference to the participants during the listening test. Participants who misunderstood the evaluation task were excluded: it includes ranking the non-expressive model higher than the other models, as well as participants with significantly lower standard deviation of grades. Examples rated by participants can be found at the following link[3].

Results of this perceptual experiments are given in Table 7. ***LA*** was ranked significantly lower than all other models, except for "Narrative" which is also modelled by the non-expressive ***LA***. Participants tend to favor ***LST$_W$*** and ***LST$_P$*** over ***GST***. Most noticeable improvements are found for "Angry", "Committed", "Enthusiastic", "Sorry" and "Narrative". Objective evaluations have shown that the LST module helps producing local behaviors that are closer to the ***GT***. Reproducing pitch variations and phrasing is critical for these styles to be perceived as natural. Note that ***GT*** exhibited relatively poor results on "Skeptical" and "Thoughtful". These styles may have been too caricatured by the speaker, which participants judged as unnatural.

[3] https://www.gipsa-lab.grenoble-inp.fr/~martin.lenglet/listening_page_LST/index.html

Table 7: *Expressive-MUSHRA results per style. Blue (resp. red) indicates that the proposed model performs better (resp. worse) than the **GST** baseline. * and ** indicates that this difference with GST is statistically significant with $p<0.05$ and $p<0.01$, respectively. **LA** = Low Anchor, **GT** = Ground-Truth.*

Style	**LA**	***GST***	***LST$_W$***	***LST$_P$***	***GT***
Angry	17.3	63.0	64.3	**68.3****	75.6
Comforting	15.4	66.2	63.5	61.5	80.5
Committed	24.9	65.1	**70.9****	68	76.4
Enthusiastic	11.6	66.2	70.0	**74.0***	86.4
Obvious	40.2	65.7	61.4	65.3	84.7
Playful	16.4	63.3	66.3	67.4	86.5
Pleading	12.3	71.3	70.1	71.2	77.9
Skeptical	36.3	47.3	50.6	46.6	63.3
Sorry	15.4	63.2	**71.1****	68.0	68.7
Surprised	14.3	78.5	75.6	73.7	85.3
Thoughtful	24.3	46.9	47.5	52.7	62.7
Narrative	64.6	63.1	**67.4***	**67.5***	69.5
Total	24.2	63.0	64.7	65.0	76.1

5. Conclusions and Discussion

In this paper, we proposed the LST module for expressive TTS which helps modeling fine-grained prosodic patterns. This module was evaluated on 12 common expressive style for French synthesis. Most promising improvements over the GST baseline are shown for "Angry", "Committed", "Enthusiastic" and "Sorry", for which more subtle prosodic variations are needed to achieve a natural behavior.

The number of tokens and training process of the LST module deserves more attention. The best results are found for styles that make use of multiple local tokens (Table 2 and Fig 2). This result was expected, since adding the same local token all along the utterance should not provide different results from an utterance-wise style bias. Constraining the LST module to maximize tokens usage should help the model showing more robust results. Additionally, the number of local tokens should be adapted to the scale of representations, e.g. allowing more various contributions for finer-grained prosodic patterns. Finding the acoustic and prosodic features encoded by the local tokens may also help understanding the acoustic similarities between styles. This analysis is left for future works.

This study reinforces the need for more elaborated evaluation paradigms for expressive speech. While style "Sorry" showed the greater amount of objective errors compared to the Ground-Truth, it was still perceived as well rendered during listening tests. Prosodic patterns followed by the Ground-Truth are not exclusive, and evaluation has to be adapted to match perceptual judgements.

We will explore cascaded LST that can be stacked to encode increasingly finer representations such as phrases, words, syllables, phonemes, etc. It would also be interesting to explore the addition of level-specific information, using pre-trained representations as BERT [28] for example.

6. Acknowledgements

This research has received funding from the BPI project THERADIA and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). This work was granted access to HPC/IDRIS under the allocation 2023-AD011011542R2 made by GENCI.

7. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[2] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv:2006.04558*, 2020.

[3] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*. IEEE, 2019, pp. 3617–3621.

[4] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[5] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[6] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[7] P. van Rijn, S. Mertes, D. Schiller, P. M. Harrison, P. Larrouy-Maestri, E. André, and N. Jacoby, "Exploring emotional prototypes in a high dimensional tts latent space," in *Interspeech*, 2021, pp. 3870–3874.

[8] P. Wu, Z. Ling, L. Liu, Y. Jiang, H. Wu, and L. Dai, "End-to-end emotional speech synthesis using style tokens and semi-supervised training," in *Asia-Pacific Signal and Information Processing Association (APSIPA)*. IEEE, 2019, pp. 623–627.

[9] A. Sorin, S. Shechtman, and R. Hoory, "Principal style components: Expressive style control and cross-speaker transfer in neural tts." in *Interspeech*, 2020, pp. 3411–3415.

[10] M. Liberman and A. Prince, "On stress and linguistic rhythm," *Linguistic inquiry*, vol. 8, no. 2, pp. 249–336, 1977.

[11] E. Selkirk, "On derived domains in sentence phonology," *Phonology*, vol. 3, pp. 371–405, 1986.

[12] K. Klapsas, N. Ellinas, J. S. Sung, H. Park, and S. Raptis, "Word-level style control for expressive, non-attentive speech synthesis," in *SPECOM*. Springer, 2021, pp. 336–347.

[13] T. Kenter, V. Wan, C.-A. Chan, R. Clark, and J. Vit, "Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3331–3340.

[14] Y. Lei, S. Yang, X. Wang, and L. Xie, "Msemotts: Multi-scale emotion transfer, prediction, and control for emotional speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 853–864, 2022.

[15] M. Lenglet, O. Perrotin, and G. Bailly, "Modélisation de la parole avec tacotron2 : Analyse acoustique et phonétique des plongements de caractère," in *Actes des Journées d'Etudes sur la Parole (JEP)*, Noirmoutiers, France, June 13-17 2022.

[16] M.-L. Hajj, M. Lenglet, O. Perrotin, and G. Bailly, "Comparing nlp solutions for the disambiguation of french heterophonic homographs for end-to-end tts systems," in *SPECOM*. Springer, 2022, pp. 265–278.

[17] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP*. IEEE, 2019, pp. 6945–6949.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[19] P.-E. Honnet, A. Lazaridis, P. N. Garner, and J. Yamagishi, "The siwis french speech synthesis database. design and recording of a high quality french database for speech synthesis," Idiap, Tech. Rep., 2017.

[20] F. Tarpin-Bernard, J. Fruitet, J.-P. Vigne, P. Constant, H. Chainay, O. Koenig, F. Ringeval, B. Bouchot, G. Bailly, F. Portet *et al.*, "Theradia: Digital therapies augmented by artificial intelligence," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2021, pp. 478–485.

[21] M. Vainio, A. Suni, and D. Aalto, "Continuous wavelet transform for analysis of speech prosody," *Tools and Resources for the Analysys of Speech Prosody (TRASP)*, 2013.

[22] C. Gobl, E. Bennett, and A. N. Chasaide, "Expressive synthesis: how crucial is voice quality?" in *Proceedings of 2002 IEEE Workshop on Speech Synthesis, 2002.* IEEE, 2002, pp. 91–94.

[23] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.

[24] M. Lenglet, O. Perrotin, and G. Bailly, "Impact of segmentation and annotation in french end-to-end synthesis," in *11th ISCA Speech Synthesis Workshop*. ISCA, 2021, pp. 13–18.

[25] ——, "Speaking rate control of end-to-end tts models by direct manipulation of the encoder's output embeddings," in *Interspeech 2022*. ISCA, 2022, pp. 11–15.

[26] I. BS, "1534-1, method for the subjective assessment of intermediate quality level of coding systems," *International Telecommunications Union, Geneva, Switzerland*, vol. 14, 2003.

[27] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webmushra—a comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, no. 1, 2018.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

Abstract — In recent years, deep neural architectures have demonstrated groundbreaking performances in various areas of speech processing, including Text-To-Speech (TTS). Models have grown in complexity to achieve almost natural synthesis, at the expense of the interpretability of the computed intermediate representations, also known as embeddings. This thesis aims to open this "black box" to explore embeddings computed by state-of-the-art TTS models. By identifying phonetic and acoustic features through linear probing, the proposed methods facilitate the understanding of how TTS structure speech information on an unsupervised manner. This work paves the way for designing more careful control mechanisms, without the need for additional data or training process. The insights uncovered through the proposed methods are employed to enhance the expressive control of TTS models. Three control mechanisms are proposed: 1) through the formatting of the textual input, 2) by imposing linear biases on the intermediate embeddings, and 3) by introducing a dedicated auxiliary module for the joint modeling of linguistic and paralinguistic information. These control methods are combined to propose a model for audiovisual generation for an embodied conversational avatar.

Keywords: Text-to-Speech, Expressive Control, Neural Network, Explainable AI, Linear Probing, Audio-Visual Synthesis

Résumé — Au cours des dernières années, les réseaux de neurones profonds ont bouleversé divers domaines du traitement de la parole, notamment en synthèse de parole à partir de texte (TTS). Ces modèles se sont complexifiés pour parvenir à une synthèse quasi naturelle, au détriment de l'interprétabilité des représentations intermédiaires calculées, également appelées plongements. Cette thèse vise à ouvrir cette "boîte noire" pour explorer les plongements calculés par les modèles TTS à l'état de l'art. En identifiant les paramètres acoustiques et phonétiques au moyen de predicteurs linéaires, les méthodes proposées facilitent la compréhension de la manière dont les TTS structurent l'information de façon non-supervisée. Ce travail ouvre la voie à la conception de mécanismes de contrôle mieux adaptés, qui ne nécessitent ni de données ni d'entraînement supplémentaires. La compréhension apportéee par les méthodes proposées a conduit à la mise en oeuvre de trois nouveaux mécanismes de contrôle expressif : 1) par le formatage de l'entrée textuelle, 2) par l'ajout de biais linéaires aux plongements intermédiaires, et 3) par l'introduction d'un module auxiliaire dédié à la modélisation conjointe de l'information linguistique et paralinguistique. Ces méthodes de contrôle sont ensuite combinées dans un modèle de génération audiovisuelle pour un avatar conversationnel incarné.

Mots clés : Synthèse Vocale, Contrôle Expressif, Réseau de neurones, Explication des réseaux d'apprentissage profond, Prédicteur linéaire, Synthèse Audio-Visuelle

GIPSA-lab, 11 Rue des Mathématiques
38400 Saint-Martin-d'Hères