MULTILINGUAL MULTISPEAKER EXPRESSIVE TEXT-TO-SPEECH SYSTEM

A PREPRINT

Cécile Macaire
IDMC, Université de Lorraine
Nancy, France

Chanoudom Prach
IDMC, Université de Lorraine
Nancy, France

Ludivine Robert
IDMC, Université de Lorraine
Nancy, France

Alena Yakavets
IDMC, Université de Lorraine
Nancy, France

January 17, 2021

ABSTRACT

The main goal of this work is from text input to be able to generate speech with expressivity for multiple languages, which are currently French and English. We used an end-to-end multilingual text-to-speech (TTS) system based on Tacotron 2 [1] enhanced by three modules; the first one for expressivity, the second one for multilingualism, and the third one for multiple speakers. The presented approach, called ERISHA [1], is experimented with 4 neural network architectures for expressivity, speaker and language encoders namely variational autoencoder (VAE), gaussian mixture variational autoencoder (GMVAE), global style token (GST) and x-vectors, the latter one has never been introduced for the given purpose. The training of these 4 models was conducted on three corpora: SIWIS for French, LJSpeech and EmoV-DB, which contains expressive speech data, for English. The generated speech was computed by WaveGlow, a Neural Vocoder. According to a subjective evaluation test, the results show that GMVAE and VAE models generates the best speech samples for French. For English, the best results are observed with the GMVAE and GST models.

***K*eywords** Text-to-Speech · Speech Synthesis · Multilingual · Expressivity

1 Introduction

Text-to-Speech synthesis is the task of producing a spectrogram (speech) from a given text input [2]. It is part of speech synthesis technologies, which include Automatic Speech Recognition (ASR) and Machine Translation (MT). As Thierry Dutoit points out in his book "An introduction to Text-to-Speech Synthesis" [3], the main purpose of such a system is to read any input text in the closest way for the human ear – the quality of speech is defined by its intelligibility (Is it clean, smooth?) and its naturalness (Does it have emotion? A good pronunciation?). The difficulty for a given language is to have a corpus with all the available words. Indeed, a language evolves with time and new words are added or modified over time. By relying on the phonetic rules and phonemes of the language, TTS synthesis allows automatic phonetization of any text. Certainly, a TTS system must model the prosody to generate human-like speech [4]. The prosody can be defined as a global entity which includes a number of phenomena in speech such as intonation, style, stress, rhythm and paralinguistic information.

The first major use of this system was for the benefit of blind people, for example, to help access to specific items in computer window or read a book [5]. In early systems the produced voice used to be quite mechanical. Nowadays technological advances have made its quality better and it became an indispensable module for human-to-computer interaction, hence its presence in many applications. These uses range from machine translation and interpretation to

[1] https://github.com/ajinkyakulkarni14/ERISHA

reading articles, or to voice assistants, customer services and call centers. Despite considerable progress in the area, generated natural speech from text still remains a challenging task [6].

This work in our project is focusing on multilingual text-to-speech system, based on a simple TTS model. It combines techniques that are already used for monolingual TTS systems with new approaches to adapt it to multilingual support. The benefits of building such systems is mainly for environments operating in several languages, such as, for example, Switzerland, with 3 official languages (French, Italian, German), or for international domains (trade, research, etc.) [7]. There exist high quality systems for major spoken languages, but that is not the case for languages with only some speakers [8]. Additional use could lie in the field of language learning or a digital assistant (same voice), operating in many languages.

Building a multilingual text-to-speech system requires a huge corpus with different characteristics in many languages. The most important is to have text and speech data in high quality.
Globalphone [9] is a useful multilingual database for multilingual speech recognition.

"The complete data corpus comprises

1. audio/speech data i.e. high-quality recordings of spoken utterances read by native speakers;
2. corresponding transcriptions;
3. pronunciation dictionaries covering the vocabulary of the transcripts and
4. baseline n-gram language models."

Multilinguality could be define in different ways, therefore it is important to agree on what we mean by it for our system. We can discuss two main approaches in the interpretation:

- Train the system in multiple languages, but for speech generation apply the selected language model to the whole text, regardless of what language it is originally written in.
- Train the system in multiple languages, detect the language in which the whole text or different parts of the text are written, and then produce the speech using the dedicated language model for each detected part.

In this work we are focusing on the first type of multilinguality. This however could serve as a base for building a system which would operate within the second definition.

The proposed work presents a Multilingual Multispeaker Expressive End-to-End TTS, based on Tacotron 2. The approach will use 4 different encoders to encode expressivity, speaker and multinlinguality. They are: variational autoencoder (VAE), gaussian mixture variational autoencoder (GMVAE), global style token (GST) and x-vectors. The last one has never been used, according to our knowledge, for this specific task.

The paper is organized as follows: Section 2 refers to the related work in the domain, Section 3 describes multilingual multispeaker expressive end-to-end TTS, Section 4 speaks about Neural Vocoder, Section 5 presents details about the utilised speech corpora and data preparation before training, Section 6 discusses the experimentation setup and the achieved results, and finally we have Sections 8 and 9 for discussion and conclusion.

2 Related Work

Initial works in this subject started on text-to-speech generation only. WaveNet (*Waveform-based statistical speech synthesis system*) [10] is a deep neural network which will directly modeling raw audio waveform from scratch, one sample at a time. Another neural model is DeepVoice 3 [11]. Char2Wav [12] an additional end-to-end model is made of a reader and a neural vocoder.

So far, several works explored training joint multilingual models in text-to-speech. Zen et al. implemented a HMM-based parametric TTS system which uses speaker and language factorization to transfer a voice to several languages [13]. In [14], Li & Zen developed a multilingual parametric neural TTS system where languages were defined by a unified input representation and shared parameters. In a more recent paper by Nachmani et al. [15], a multilingual neural TTS model were introduced to support voice cloning across languages. They used specific speaker encoders and optimized the loss to preserve speaker identity. However, the generated speech quality was not comparable with the recent neural TTS systems based on Tacotron 2.

To briefly describe Tacotron 2 [6], developed by Google, it is a recurrent sequence-to-sequence feature prediction network. It directly maps an input text to mel-spectrogram. Different benefits from Tacotron 2 [6] are investigate in the following studies. In [16], Chen et al. used cross-lingual transfer learning in end-to-end TTS system based on

Tacotron 2 for low-resource languages. They found out that their proposed approach enables to generate more natural speech. Prakash et al. [17] tried knowledge sharing for Indian Languages. In more details, they explored character and phone-based text representations for training Indian TTS in an end-to-end framework. Voice cloning techniques were explored by Zang et al. [18] in order to have a model which transfers voices across languages. In this study, they proposed the VAE to an end-to-end TTS system to model latent representation of speaking style. They showed that it outperformed the GST model. In [19] they implemented code-switched TTS systems useful for alternate languages.

In this work, we also worked with expressivity. Building expressive speech synthesis systems kept rising in the research community. Previous TTS models were generated "average" prosodic style, leading to problematic issues such as not considering variation in pitch (pitch declines at the end of a sentence) and therefore problematic for expressive datasets [4]. In [4], they proposed a novel model architecture based on Tacotron [1], GST, for expressive long-form synthesis to control and transfer style. In [20], an extension to Tacotron architecture was implemented to transfer prosody. Finally, in [21], they introduced the Text-Predicted Global Style Token (TP-GST) architecture which predicts stylistic renderings from text.

3 Model architecture

The multilingual multispeaker expressive TTS model is based on Tacotron 2 [6] which uses an attention-based sequence-to-sequence model to generate a sequence of log-mel spectrogram frames based on an input text sequence. In order to work with multiple languages, speakers and expressivity, we augmented the base Tacotron 2 [6] by adding a language encoder, a speaker encoder and an expressivity encoder. More details about the architecture of Tacotron 2 are presented in [6].

3.1 Generalized approach

The proposed architecture takes input as text, which is then converted to a sequence of phonemes. The different modules are described below and can be seen in the Figure 1.

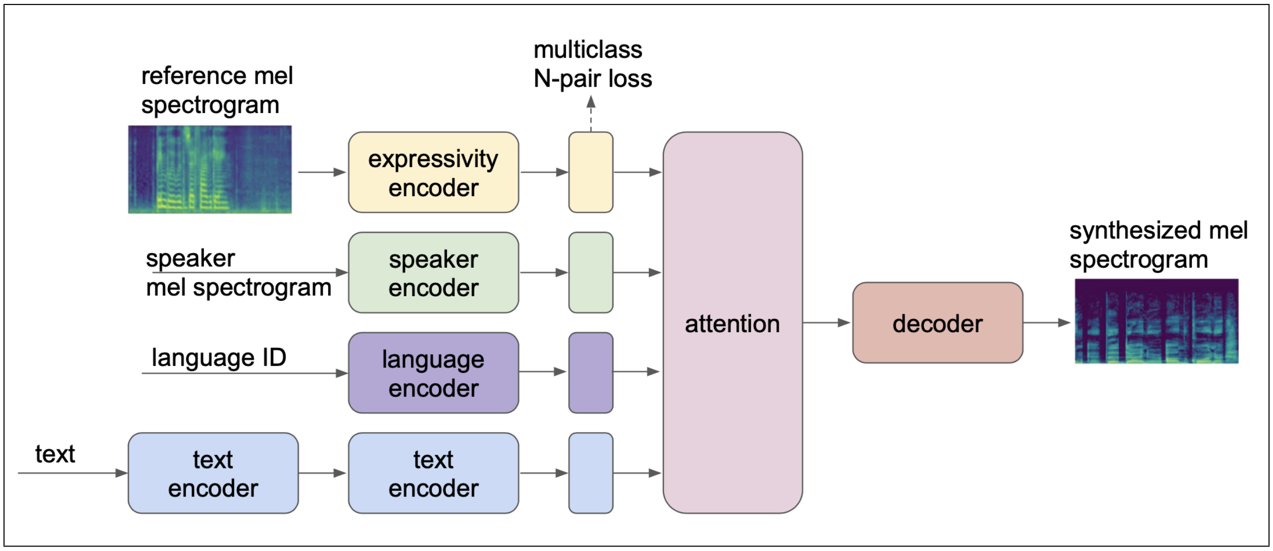


Figure 1: End-to-end Multilingual Multispeaker Expressive Text-to-Speech system.

Text Encoder
The main goal of the text encoder is to produce a latent representation of the input text. It gets through several steps. First, input characters are converted into a character embedding (512-dimension). It is then passed trough a stack of 3 convolutional layers (for each, 512 filters of shape 5 x 1). The filters defined by the text encoder take into account 5 characters, and it is followed by a batch Normalization and ReLU activations. The input character sequence is modeled into N-grams thanks to the convolutional layers. The last output generated by the convolutional layers is then converted to z_t as a latent representation of text by using a single bi-directional LSTM recurrent neural network layer of 512 units.

Expressivity, speaker and language encoders are implemented with 4 different types of neural network architecture that will be explained in Section 3.2.

Expressivity Encoder
The idea behind this module is to generate a latent representation of the emotion from a given mel spectrogram. The representation is encoded as an expressive embedding z_e.

Speaker Encoder
Speaker encoder is used to synthesise the speaker characteristics from a reference speech signal, in this case a mel spectrogram. The speaker encoder is able to capture the properties of different speakers, given a short speech signal without taking into account the phonetic content. The generated output is a non linear fixed-dimensional embedding vector z_s which maps the speaker indexes.

Language Encoder
To deal with multilinguality, a language encoder was introduced in this novel architecture. It selects language identifier to create a new embedding z_l. For this study, we only used two languages, therefore two language IDs: one for English and one for French.

Attention module & Decoder
The 4 different encoders provide latent representations (embeddings) for expressivity, language, speaker and text. The embedding vectors z_t, z_e, z_s and z_l are concatenated. The full concatenated encoded sequence will then be given to the attention module as a fixed-length context vector for each decoder output step [6]. Location-sensitive attention mechanism is used [22]. This specific attention module takes from the previous decoder time steps the cumulative attention weights as an additional feature. 128-dimensional hidden representations are generated by taking the inputs and features to compute the attention probabilities. Attention mechanism will enable to learn the alignment between the sequence of phonemes and desired mel spectrogram.

The decoder network is an autoregressive recurrent neural network which includes a pre-net, BLSTM and convolutional layer based post-net. It will predict the mel spectrogram frame by frame from the encoded input sequence. To be more precise, the pre-net which is made of 2 fully connected layers of 256 hidden ReLU units takes the prediction from the previous steps. It plays a central role in the attention mechanism because it acts as an information bottleneck. A stack of 2 BLSTM layers with 1024 units are then used on the concatenated pre-net output and attention context vector and then projected through a linear transform to define the target mel spectrogram frame. Finally, 5-convolutional layers based post-net takes the output from pre-net to improve the overall reconstruction performance of mel spectrogram [23].

Multiclass N-pair loss
To make sure the expressivity is transferred correctly, we need to have tightly bounded, clustered representation of emotion latent variables. One of the possible ways to help with that is using triplet loss. It is a loss function for machine learning algorithms where a baseline input is compared to a positive input and a negative input. The distance from the baseline input to the positive input is minimized, and to the negative input – maximized [24]. By positive examples we mean latent variables from the same expressive class, and other different classes by negative. In our context with multiple emotion classes, this method does not appear to be optimal. In order to enhance expressivity representation, our approach is utilizing a learning framework with multiclass N-pair loss. It allows joint comparison among more than one negative examples and is also more efficient computationally. Learning to identify from multiple negative examples, multiclass N-pair loss is showing higher performance than triplet loss or contrastive loss. In our approach it successfully reduces the distance between latent variables of the same emotion class, increasing the inter-cluster distance from N1 negative samples (z^-) and decreasing the intra-cluster distance between positive samples (z^+) and training examples.

In the training phase, the model needs to reduce the multi- class N-pair loss function according to the following formula:

$$\log(1 + \sum_{i=1}^{N-1} \exp{(z_e^T z_i^- - z_e^T z^+)})$$

3.2 Encoders

As we mentioned at the beginning, we experimented with 4 neural network architectures for the implementation of expressivity, speaker and language encoders namely GST, VAE, GMVAE and x-vectors.

3.2.1 GST

GST stands for "Global Style Tokens". It is a bank of embeddings that are trained together within Tacotron. When trained on expressive speech data, with no explicit prosodic labels, a GST model learns to model acoustic expressiveness independently of the text content and yields interpretable embeddings that can be used to control and transfer style.

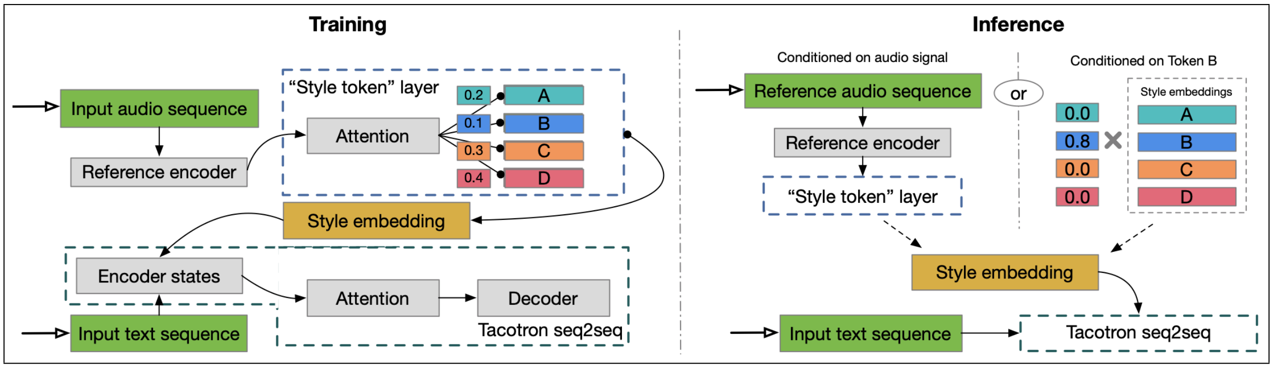


Figure 2: GST model diagram [4].

Figure 2 explains the general architecture of the GST model.

The training is splitted into 4 different modules:

1. The *reference encoder*, proposed in [20], encodes into a fixed-length vector the prosody of the speech signal (reference embedding). It is made made up of a convolutional stack with a Recurrent Neural Network (RNN). The input of this module is a log-mel spectrogram.

2. The generated reference embedding is the input to the *attention module* which is passed through a "style token" layer. The purpose here is to learn the similarities between the reference embedding and the tokens from a randomized set of embeddings, namely tokens embeddings or global style tokens (GSTs) are the same for all the training sequences.

3. As we see in the Figure 2, a *style embedding* is defined after the attention module. Style embedding refers to the weighted sum of the token embeddings (GSTs). Indeed, the attention module output is a set of weights which represent each style token weight attributed to the reference embedding. the style embedding is proceeded by the text encoder for conditioning.

4. *Tacotron 2 architecture* is trained in parallel with the style token layer (cf. Tacotron seq2seq in Figure 2).

The inference mode to synthesize text with its speaking style can be of two ways. The first one, on the right-hand side of the inference module in Figure 2 is based on the fact that the text encoder can only select certain tokens to control the style without a reference signal. The second approach is to condition the reference encoder on a different audio signal for style transfer (see left-hand side of the inference module in Figure 2).

In the train phase, the train target log-mel spectrogram is submitted to the reference encoder, and then is processed in the style token layer. The resulting style embedding conditions text encoder states in Tacotron. In the inference phase, it is either a reference audio sequence, conditioned on audio signal, that is used to synthesize speech with a desired expressive style, or, alternatively, the reference encoder is skipped and speech synthesis is controlled directly via the learned interpretable tokens.

3.2.2 VAE

VAE is a deep generative model and had succeeded in the tasks of text generation, image generation and speech generation [18]. It corresponds to the second encoder architecture and is used to learn the latent representation of speaking styles. It was first introduced to Tacotron 2 in "Learning latent representations for style control and transfer in end-to-end speech synthesis" [18]. The main idea is that the speaking style is easily controlled by manipulating disentangled latent variable or variational inference from a reference audio. The generated speech style can then be defined. VAE generates speech with different speaking styles thanks to the direct sampling on prior of latent distribution, useful when new data is introduced.

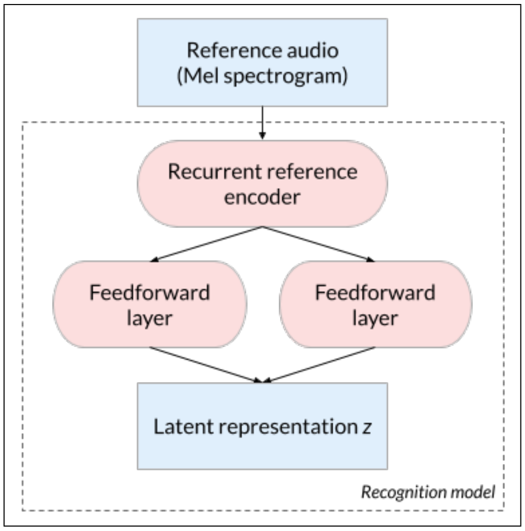


Figure 3: VAE encoder architecture [18].

VAE architecture (recognition model) consists of a recurrent reference encoder to encode the reference speech to latent representation (a fixed-length short vector) and two fully connected feedforward layers to generate mean and standard deviation of latent variable z (cf. Figure 3). Training VAE encoder is followed by a Kullback Leibler (KL) annealing problem, meaning that the KL loss collapsed (dropped to 0) before giving a proper latent representation [18]. The solution here was to introduce a variable weight closed to 0 and multiplied it to KL loss every training steps.

3.2.3 GMVAE

GMVAE-Tacotron models latent attributes using a mixture distribution, which allows automatic discovery of latent attribute clusters. Its structures make it more simple and easier to interpret the underlying latent space. The model learns an interpretable and disentangled latent representation to enable fine-grained control of latent attributes and provides a systematic sampling scheme for them. If speaker labels are available, we demonstrate an extension of the model that learns a continuous space that captures speaker attributes, along with an inference model which enables one-shot learning of speaker attributes from unseen reference utterances.

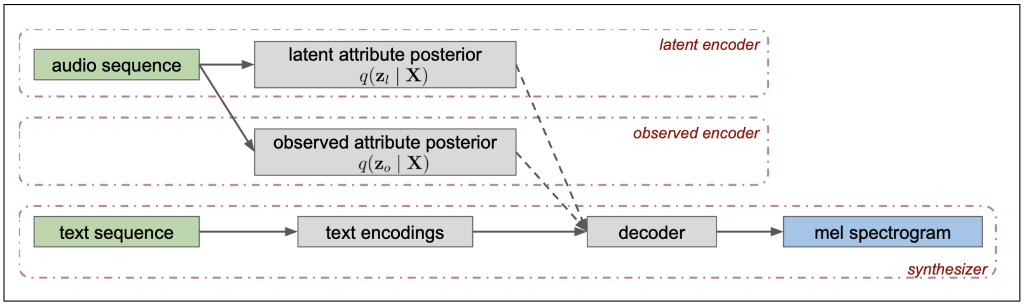

audio sequence
latent attribute posterior
$q(\mathbf{z}_l \mid \mathbf{X})$
latent encoder
observed attribute posterior
$q(\mathbf{z}_o \mid \mathbf{X})$
observed encoder
text sequence
text encodings
decoder
mel spectrogram
synthesizer

Figure 4: Training configuration of the GMVAE-Tacotron model [25].

Dashed lines denotes sampling. The model consists of three modules: a synthesizer, a latent encoder, and an observed encoder.

3.2.4 X-VECTORS

The last and new one x-vectors follows i-vectors which have been shown to be effective in speaker verification and recognition systems [26]. x-vectors is a hidden layer extracted feature vector. It is based on feed-forward deep neural network (DNN) embeddings, which employs a multiple layered DNN architecture (with fully connected layers) with different temporal context at each layer (called 'frames').

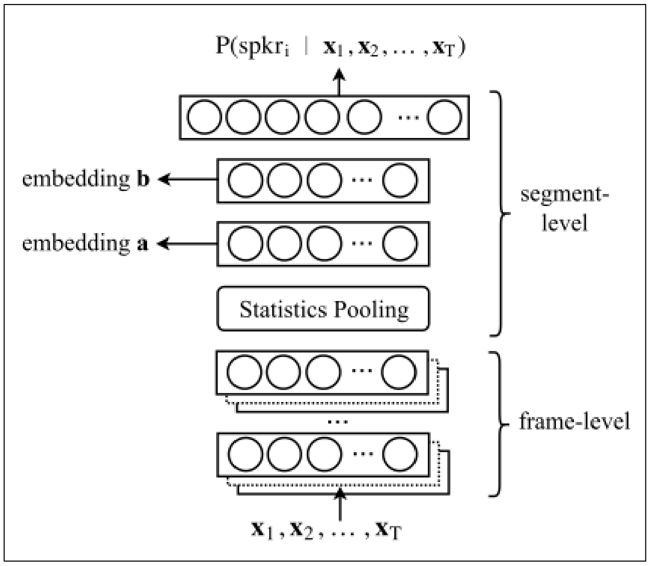

P(spkr$_i$ | $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$)
embedding **b**
embedding **a**
Statistics Pooling
segment-level
frame-level
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$

Figure 5: x-vectors DNN embedding architecture [27].

As we see in Figure 5, the layers first operate on speech frames (frame-level). Then, a statistics pooling layer aggregates the frame-level embeddings, followed by layers operating at the segment-level. The final step is a softmax output layer. There are 5 layers at the frame-level based on a time-delay architecture. The time-delay architecture works as follows: at t time, layer 1 splices the frames at $t-2, t-1, t, t+1, t+2$. Then, layers 2 and 3 splice the output of the previous layer at times $t-2, t, t+2$ and $t-3, t, t+3$, respectively. Finally, layers 4 and 5 process but without any temporal information. The input of the statistics pooling layer is the output of the last frame-level layer. It calculates the mean and standard deviation. The embeddings are computed with two hidden layers which concatenate the previous segment-level statistics.

4 Neural Vocoder

WaveGlow [28] neural vocoder was incorporated to generate speech waveform from mel-spectrograms. WaveGlow uses the knowledge of Glow [29] and WaveNet [10], neural network based models that can synthesize speech without auto-regression. Previous approaches were initially based on autoregressive neural network, meaning that the audio samples are generated on previous one. Despite the fact that these models are simple and fast to train, they tend to generate poor quality speech. As for Glow [29] and WaveNet [10], these neural network based models can synthesize audio at more than 500kHz on a GPU but are difficult to train and implement. WaveGlow [28] overcomes the previous studies by proposing a flow-based network which uses only a single network and is easy and fast to train. It consumes the mel spectrograms to generate speech.

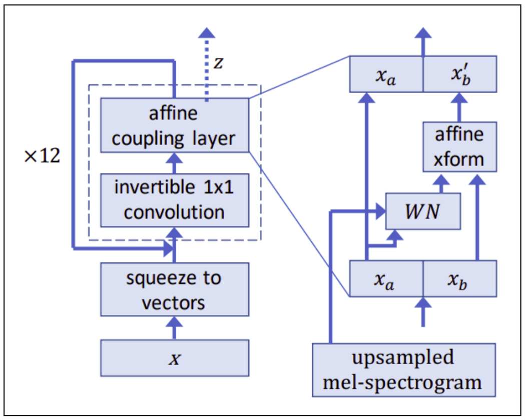


Figure 6: WaveGlow architecture [28].

WaveGlow is following the corresponding architecture displayed in Figure 6. Speech samples are used as vectors to pass through the network, the step is called "squeeze" operation. Vectors are then processed by several modules as they are called "steps of flow" (invertible 1 × 1 convolution and affine coupling layer). The output z is then generated by concatenating the final vectors with the previous output channels.

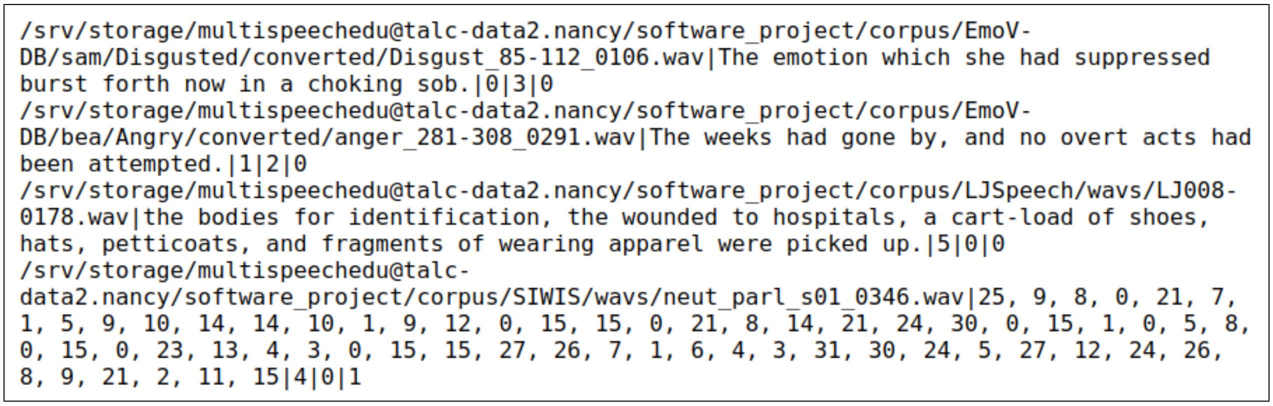
5 Data Preparation

In this project, we used 3 freely available speech synthesis corpora for implementing end-to-end multilingual multispeaker expressive TTS system. The speech corpora are in French and English: SIWIS [30], EmoV-DB [2] [31] and LJSpeech [32] respectively. The French corpus contains a great quality of french speech recordings and text files used to build a TTS system. In total, 9750 utterances from numerous sources such as novels, read by a professional female neutral voice speaker; and also took from parliament debates in a neutral form (more than 10 hours of speech). The first English corpus, LJSpeech is 13,100 short audio clips. A single speaker reads passages from 7 non-fiction books. Each clip lasts from 1 to 10 seconds and has its own transcription (approximately 24 hours of speech in total). The second English speech corpus, EmoV-DB, consists of 5 emotions namely: amusement, anger, sleepiness, disgust and neutral, where two male and two female native speakers read sentences from books, taken from CMU-arctic database [3]. Table 5 bellow summarizes the datasets.

Languages	**French**	**English**	
Corpus	SIWIS	LJSpeech	EmoV-DB
Size	$>$ 10 hours	24 hours	$\approx$ 10 hours
Number of speakers	1	1	4
Expressivity	No	No	Yes

Table 1: Description of the databases used for model training.

As input features to end-to-end TTS system, we created a filelist containing the path of audio speech, a sequence of phonemes for French (extracted from the text with a grapheme-to-phoneme conversion SOJA-TTS tool developed internally in Multispeech team) and a sequence of text for English, a speaker identifier, an emotion identifier and a language identifier (cf. Figure 7). In total, over 30000 speech utterances are present. The whole set of speech corpus is split into train, validation, and test sets in 90:5:5 ratio respectively.



```
/srv/storage/multispeechedu@talc-data2.nancy/software_project/corpus/EmoV-
DB/sam/Disgusted/converted/Disgust_85-112_0106.wav|The emotion which she had suppressed
burst forth now in a choking sob.|0|3|0
/srv/storage/multispeechedu@talc-data2.nancy/software_project/corpus/EmoV-
DB/bea/Angry/converted/anger_281-308_0291.wav|The weeks had gone by, and no overt acts had
been attempted.|1|2|0
/srv/storage/multispeechedu@talc-data2.nancy/software_project/corpus/LJSpeech/wavs/LJ008-
0178.wav|the bodies for identification, the wounded to hospitals, a cart-load of shoes,
hats, petticoats, and fragments of wearing apparel were picked up.|5|0|0
/srv/storage/multispeechedu@talc-
data2.nancy/software_project/corpus/SIWIS/wavs/neut_parl_s01_0346.wav|25, 9, 8, 0, 21, 7,
1, 5, 9, 10, 14, 14, 10, 1, 9, 12, 0, 15, 15, 0, 21, 8, 14, 21, 24, 30, 0, 15, 1, 0, 5, 8,
0, 15, 0, 23, 13, 4, 3, 0, 15, 15, 27, 26, 7, 1, 6, 4, 3, 31, 30, 24, 5, 27, 12, 24, 26,
8, 9, 21, 2, 11, 15|4|0|1
```

Figure 7: Extract from the train filelist.

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

[3] http://www.festvox.org/cmu_arctic/

6 Experimentation

As mentioned in Section 3, we trained 4 models (GST, VAE, GMVAE and x-vectors), one for each encoder which encompasses expressivity, speaker and language.

To train our models we were using Grid5000 [4] [33], which is a large-scale and flexible testbed for experiment-driven research in most of the fields of computer science and high computing in Cloud, Big Data and AI. The server enabled us to successfully train our models, which demand a lot of memory usage, storage and powerful performance – exceeding by far the capabilities of a regular computer.

To train the end-to-end TTS system, the same parameters were used from Tacotron 2 architecture. As for the encoders, the parameters were also shared between them. 128 dimensional latent variable of expressivity were used for the 4 encoders. Number of symbols (i.e. phonemes) was set to 38. It corresponds to the number of phonemes observed in both languages (French and English). The embedding dimension was equal to 256. We used a sampling rate of 22050 Hz, a filter size and a window size of 1024, a hop length of 256. 80 Mel filters were used to extract mel spectrograms. The Kullback Leibler (KL) problem that we mentioned in section 3 needed the set of a weight of 0.0001 in every 200 steps. Moreover, to fine-tune the multiclass N-pair loss, we set the weight to 0 till 150k training steps and then increased by 0.001 after every 200 steps. Each encoder model was trained for 500 epochs. Table 2 displays the hyperparameters that were used. WaveGlow was incorporated as the neural vocoder to generate speech from mel-spectrograms (see Section 4). Finally, the generated speech were evaluated by comparing the performance of the 4 models GST, VAE, GMVAE, x-vectors (see Section 7).

Parameters	**Value**
Epoch	500
Learning rate	0.001
Weight decay	0.000001
Convolution Layer 1	Kernel Size = 3
Batch Size	1

Table 2: Hyperparameters shared by the models.

7 Results

To estimate the performance of our models we considered two types of approaches: an objective evaluation, analysing the values of certain acoustic parameters, and a subjective evaluation, conducting an opinion poll among participants to rate the synthesized samples. Knowing that subjective evaluation is deemed to be more relevant for assessing synthesized speech perceived quality, we proceeded with this type of evaluation.

7.1 Subjective evaluation

To conduct subjective evaluation and get feedback from the participants we developed a dynamic website, that allowed selecting data either in French or in English, according to the language support of our models. The website can be found at the following link: `https://evaluationtts.herokuapp.com/index.php`.

We evaluated the end-to-end (E2E) multilingual multispeaker expressive TTS system utilizing Mean Opinion Score (MOS) [34] metric. This unified measurement was to rate such parameters as intelligibility, naturalness and the quality of the speech utterance. We decided to use absolute category ranking which ranges from 1 to 5. Every listener had to choose the score for the synthesized speech utterance from with the corresponding values:

- "1" – poor
- "2" – fair
- "3" – good
- "4" – very good
- "5" – excellent

[4] `https://www.grid5000.fr/w/Grid5000:Home`

The total number of participants for the subjective evaluation was 30. They distributed as 14 people for English and 16 for French. The majority of them were knowledgeable in the domains of NLP or Linguistics.

For a participant to be able to evaluate if the system generated the audio correctly from the given text, we have decided to display the text next to the audio for reference. We were aware about the risk that it might bias the participants to give a slightly higher grade, as text would help with understanding each phrase better. However for us it was important that the system would not mispronounce any word in a way that would theoretically exist and thus would not be spotted as an error.

The evaluation per language contains 10 randomly selected synthesized speech samples for each model: targeting 2 per speaker from the corpora on which the models were trained.

In addition, to evaluating speech samples synthesized by the TTS model, we originally intended to also perform subjective scoring of the expressivity transfer for a given set of emotions. The achieved quality of the produced results however was not quite up to the mark. Consequently, the decision was that this module needs further fine-tuning before being submitted for subjective evaluation.

The results of this evaluation can be found in Tables 3 for French, 4 for English, and 5 for combined results.

It is important to note, that for synthesizing English samples two very different types of corpora were used: one containing highly emotional speech, the other one normal. We dropped the evaluation of emotional contour transfer, however we still wanted to conduct the assessment for more than one speaker, we have as well used some samples from the Emo-DV corpus, but only with the 'neutral' emotion. Knowing that the corpus size is almost 2.5 times smaller compared to LJSpeech, and also needs to be divided by the number of speakers and the number of emotions, the total available training samples for neutral emotions ended up being not particularly big. For certain speakers the intelligibility of produced speech was quite low, so we skipped them for the evaluation and instead added more samples from the LJSpeech corpus. It was also interesting to see the difference in the average scores between these two corpora, therefore we calculated the average MOS for them separately in addition to the total average.

Model	**Avg MOS**	**Number of samples**
x-vectors	2.59	10
GMVAE	**2.83**	10
VAE	**2.73**	10
GST	2.59	10

Table 3: MOS score for French samples evaluation per different trained model.

For the French synthesized speech we can observe that the average MOS for all 4 models was rated below 3. The highest results were achieved by GMVAE and VAE models with the MOS of 2.83 and 2.73 respectively. This does not correspond with the results achieved with the base ERISHA library [23], where GST had obtained better results than VAE by 0.25 in term of average MOS.

Model	**Corpus**	**Avg MOS**	**Number of samples**
x-vectors	Emo	**2.37**	5
GMVAE	Emo	2.26	7
VAE	Emo	2.17	6
GST	Emo	**2.57**	6
x-vectors	LJ	2.49	5
GMVAE	LJ	**3**	3
VAE	LJ	2.48	4
GST	LJ	**2.95**	4
x-vectors	LJ+Emo	2.43	10
GMVAE	LJ+Emo	**2.56**	10
VAE	LJ+Emo	2.26	10
GST	LJ+Emo	**2.72**	10

Table 4: MOS score for English samples evaluation per different trained model.

ERISHA library, based on which we trained our models, hasn't been used with English data before, so we have no formal reference data for our corpora. In the achieved results we can observe that the highest performing models are GST and GMVAE, with the average MOS score of 2.72 and 2.56 respectively. This is only partially similar to the behavior we observed for the French data, where it is GMVAE and VAE that achieved better results.

We have combined the results from both languages into a global Table 5 with averaged MOS values.

Model	**Avg MOS**
x-vectors	2.51
GMVAE	**2.70**
VAE	2.50
GST	**2.66**

Table 5: Combined MOS score for French and English samples evaluation per different trained model.

In the averaged results, the mean score for GST and GMVAE models turned out to be rated the highest, though still faring below the "good" midpoint grade. We should note though that the number of speakers is different in between the corpora, as well as the corpora size, which might have affected the results and made them dissimilar.

8 Discussion

There are several things that we discussed a lot about, while working on this project.

First of all, we would like to emphasize such a well-known thing as the importance of good and well processed data. Even if the corpora that we used is widely known, several issues were encountered. The first one was during the generation of the filelist from the data. For some speech utterances, the transcription was missing. One speech file from the French corpus SIWIS lasted 10 minutes and could not be properly processed, because the TTS system developed for this project only takes into account small speech files (<1 min). An additional problem here is that to detect these issues, we sometimes had to train the models for several hours and then to restart, which summed up to a significant time lost in the end. In total, we lost almost a week to spot and eliminate all issue in the filelist and to be able to train the models properly.

It was an interesting experience to be able to use an external powerful computer for training – Grid500. It allows to train models in passive mode, which is a big gain, but requires some adaptation. We have encountered certain difficulties due to administrative and permissions issues regarding Grid5000. Theoretically it is possible to train the model for 168 hours straight in a passive mode. But because of our student accounts, the training was only possible for 24 hours and had to be relaunched every day. We had frequent checkpoints to save the training progress, however it still caused some loss in time due to overlaps or unsaved progress due to a timeout before a checkpoint was reached.

To train ERISHA, some requirements need to be met and specific libraries, such as torch, to be installed. Their installation in the Grid5000 repository and resolving all the conflicts took us a quite bit of time too.

We see that there are a lot of developments in the recent years in the field of text-to-speech generation. It is fascinating to see what the systems are capable of! There is of course still a lot of room for improvements, to make speech more natural and to cover more languages, which is very important globally.

For us it was interesting to see the architecture of different models that we trained our system on, what type of data they take as input and how they process it to achieve the desired results. Of course, without digging deep into each model it is hard to pinpoint why one model performs better that the other with particular settings. It would be interesting to play with the training parameters to see how the results can be improved.

Finally, we currently only performed subjective evaluation of the speech in terms of intelligibility, naturalness, quality, but we omitted the emotions transfers module due to low results. For the future it would be nice to optimize training parameters to achieve better results and submit them for a subjective evaluation as well.

There was an existential question that came into our mind while looking into the automated emotion transfer. The question was whether emotional contours are universal or culturally biased, and whether it depends on such factors as primal (instinctive) emotions, like fear, or more socially advanced ones, like disgust. To be able to monitor this, it would require a big set of languages from different cultural heritages to properly compare the data.

9 Conclusion

In conclusion we experimented extensions of Tacotron 2, which include multiple speakers, expressiveness and languages. We trained 4 models (GST, VAE, GMVAE, x-vectors) on low resources. The results of the subjective evaluation test show globally that GMVAE achieves the highest performance among the given models. For future work, we would like to add more languages. In this work, we only included 3 corpora: 2 for English and 1 for French. We also would like to add more data per language. The next step will take into account the expressivity of the generated speech. Finally, we would rate the speech samples by more people and perform evaluation, based on objective parameters (e.g. MCD, F0 RMSE, VUV).

10 Acknowledgement

We would like to thank Ajinkya Kulkarni for his support throughout the project and the availability, the time spent explaining the details of the tasks and solving the problems encountered during the training phases. We also thank the people who took part of the evaluation. Finally, experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations.

References

[1] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[2] Zhizheng Wu, Oliver Watts, and Simon King. Merlin: An open source neural network speech synthesis system. In *9th ISCA Speech Synthesis Workshop*, pages 202–207, 2016.

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

[4] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*, 2018.

[5] Paul Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

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

[7] Herve Bourlard, John Dines, Mathew Magimai-Doss, Philip N Garner, David Imseng, Petr Motlicek, Hui Liang, Lakshmi Saheer, and Fabio Valente. Current trends in multilingual speech processing. *Sadhana*, 36(5):885–915, 2011.

[8] Johannes A Louw. Speect: a multilingual text-to-speech system. 2008.

[9] Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe. Globalphone: A multilingual text & speech database in 20 languages. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8126–8130. IEEE, 2013.

[10] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[11] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*, 2017.

[12] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis. 2017.

[13] Heiga Zen, Norbert Braunschweiler, Sabine Buchholz, Mark JF Gales, Kate Knill, Sacha Krstulovic, and Javier Latorre. Statistical parametric speech synthesis based on speaker and language factorization. *IEEE transactions on audio, speech, and language processing*, 20(6):1713–1724, 2012.

[14] Bo Li and Heiga Zen. Multi-language multi-speaker acoustic modeling for lstm-rnn based statistical parametric speech synthesis. 2016.

[15] Eliya Nachmani and Lior Wolf. Unsupervised polyglot text-to-speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7055–7059. IEEE, 2019.

[16] Yuan-Jui Chen, Tao Tu, Cheng-Chieh Yeh, and Hung-Yi Lee. End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning. In *Interspeech*, pages 2075–2079, 2019.

[17] Anusha Prakash, Anju Leela Thomas, S Umesh, and Hema A Murthy. Building multilingual end-to-end speech synthesisers for indian languages. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 194–199.

[18] Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua 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)*, pages 6945–6949. IEEE, 2019.

[19] Yuewen Cao, Xixin Wu, Songxiang Liu, Jianwei Yu, Xu Li, Zhiyong Wu, Xunying Liu, and Helen Meng. End-to-end code-switched tts with mix of monolingual recordings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6935–6939. IEEE, 2019.

[20] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv preprint arXiv:1803.09047*, 2018.

[21] Daisy Stanton, Yuxuan Wang, and RJ Skerry-Ryan. Predicting expressive speaking style from text in end-to-end speech synthesis. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 595–602. IEEE, 2018.

[22] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. 2015.

[23] Ajinkya Kulkarni, Vincent Colotte, and Denis Jouvet. Improving latent representation for end to end multispeaker expressive text to speech system. 2020.

[24] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 1857–1865. Curran Associates, Inc., 2016.

[25] Wei-Ning Hsu, Yu Zhang, Ron Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, Patrick Nguyen, and Ruoming Pang. Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*, 2019.

[26] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[27] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, pages 999–1003, 2017.

[28] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[29] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018.

[30] Junichi Yamagishi, Pierre-Edouard Honnet, Philip Garner, Alexandros Lazaridis, et al. The siwis french speech synthesis database. 2017.

[31] Adaeze Adigwe, Noé Tits, Kevin El Haddad, Sarah Ostadabbas, and Thierry Dutoit. The emotional voices database: Towards controlling the emotion dimension in voice generation systems. *arXiv preprint arXiv:1806.09514*, 2018.

[32] Keith Ito and Linda Johnson. The lj speech dataset. 2017. *URL https://keithito. com/LJ-Speech-Dataset*, 2017.

[33] Daniel Balouek, Alexandra Carpen Amarie, Ghislain Charrier, Frédéric Desprez, Emmanuel Jeannot, Emmanuel Jeanvoine, Adrien Lèbre, David Margery, Nicolas Niclausse, Lucas Nussbaum, Olivier Richard, Christian Pérez, Flavien Quesnel, Cyril Rohr, and Luc Sarzyniec. Adding virtualization capabilities to the Grid'5000 testbed. In Ivan I. Ivanov, Marten van Sinderen, Frank Leymann, and Tony Shan, editors, *Cloud Computing and Services Science*, volume 367 of *Communications in Computer and Information Science*, pages 3–20. Springer International Publishing, 2013.

[34] Robert Streijl, Stefan Winkler, and David Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22:213–227, 03 2016.