EMOTRON: an Expressive Text-to-Speech

Cristian Regna[1], Licia Sbattella[2], Vincenzo Scotti[3],
Alexander Sukhov[4] and Roberto Tedesco[5]

[2]https://orcid.org/0000-0001-5344-5976

[3]https://orcid.org/0000-0002-8765-604X

[5]https://orcid.org/0000-0002-2830-4247

DEIB, Politecnico di Milano
Via Golgi 42, 20133, Milano (MI), Italy
[1]cristian.regna@mail.polimi.it, [2]licia.sbattella@polimi.it,
[3]vincenzo.scotti@polimi.it, [4]alexander.sukhov@polimi.it,
[5]roberto.tedesco@polimi.it

Abstract

The introduction of end-to-end deep learning architectures for spectrogram prediction in *Text-to-Speech* (TTS) synthesis significantly pushed forward the state of the art. More recent works, deal with conditioning the synthesis model upon different aspects, such as speaking style. With this work, we focus specifically on the conditioned generation of emotional speech. To introduce control over the expressed emotion, augmenting a *Tacotron* spectrogram predictor with an *emotional transfer module*. The resulting architecture, namely *EMOTRON*, was trained with a combination of a spectrogram regression loss, to enforce synthesis, and an emotional classification style loss, to induce the conditioning. The obtained model is able to generate a very fluent and expressive voice, given the input sentence to be pronounced and the target emotion, as confirmed by a human evaluation we conducted. When compared to similar systems, EMOTRON obtains comparable results in terms of speech quality and consistently better results in terms of emotion clarity.

Keywords: Natural Language Processing, Deep-Learning, Text-to-Speech, Emotion conditioning.

1. Introduction

Text-to-Speech (TTS) synthesis (or, simply, speech synthesis) is the task of synthesising a waveform uttering a given piece of text [1]. Like many other areas of the Artificial Intelligence (AI) field, Natural Language Processing (NLP) has been pervasively affected by deep learning. The subsequent development of Neural TTS systems (i.e., neural networks for speech synthesis) has dramatically pushed forward the state-of-the-art [2].

In the latest years these neural network-based models evolved significantly, introducing neural vocoders to significantly improve the quality of the synthesised speech [2]. They also introduced the possibility of conditioning the synthesised speech on different aspects, like the speaker's vocal timbre} or the prosodic style [2]. This latter aspect enables the TTS to be significantly expressive when uttering a sentence.

With this chapter, we introduce EMOTRON, a TTS able to condition the synthesised speech on a given emotion[1]. The idea is to control the emotion expressed by the utterance by providing such emotion as an additional input to the neural network, during the synthesis process. During training, the network is updated to minimise speech synthesis and perceived emotion losses. In this way, we have the network learn to control the prosody (i.e., the voice rhythm, stress, and intonation) necessary to deliver the emotional information through the uttered speech.

To assess the quality of our model, we conducted an evaluation based on human opinions. Listeners were asked to compare EMOTRON synthesised speech to real clips and to clips synthesised by a reference Tacotron 2 TTS we trained as a baseline. Compared to natural clips, those synthesised by EMOTRON and the baseline were always inferior. However, when compared to the baseline TTS, our results were slightly worse in terms of clarity of speech and clearly better in terms of perceived emotion.

We organise the rest of this chapter according to the following structure. In Section 2 we present the current approaches in the Deep Learning-based TTS development. In Section 3 we introduce our architecture for emotional TTS. In Section 4 we explain how EMOTRON is trained and used at inference. In Section 5 we present the corpora we employed to train our model. In Section 6 we describe the evaluation approach we followed to assess the quality of our model. In Section 7 we present and comment the results of the evaluation of our TTS. In Section 8 we summarise our work and provide hints about possible future extensions.

2. Related Works

In this section we outline the main aspects concerning neural TTS synthesis models and we provide details concerning the control of speech style in such systems.

2.1. Text-to-Speech synthesis with neural networks

Deep neural networks enabled many new possibilities in developing TTS systems. In this chapter, we focus on approaches based on acoustic models [2]. TTS developed with this architecture are divided into two main components: acoustic model (also called spectrogram predictor) and vocode [1] [2]. The former component takes care of converting a sequence of

[1] The implementation is available at https://github.com/Sashorg/Emotional_TTS-master

graphemes[2] or phonemes[3] into a Mel-spectrogram; the latter component takes care of converting the Mel-spectrogram into a raw waveform, concluding the synthesis process.

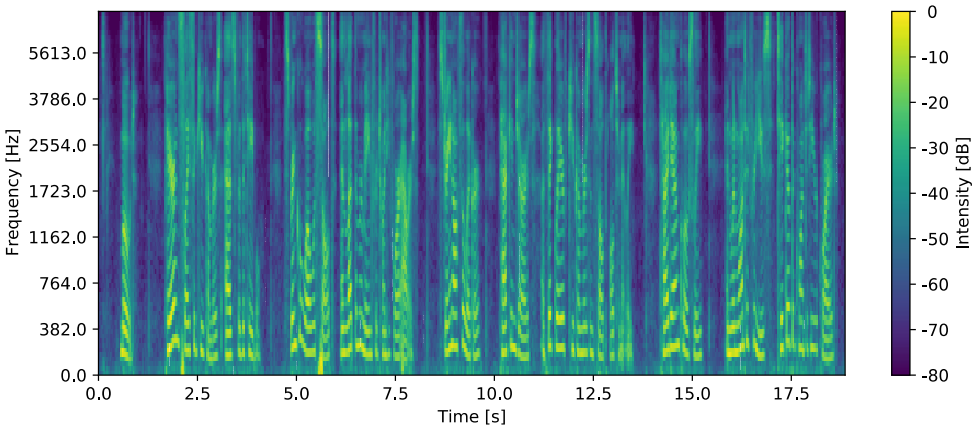

Figure 2.1. Visualisation of a Mel-spectrogram.

A spectrogram is a visual way of representing a signal strength over time, at various frequencies. If such frequencies are passed through a mel filter (a mathematical model trying to approximate the non-linear human sensitivity to various frequencies), the result is the Mel-spectrogram. Figure 1 shows an example; horizontal axis shows time, vertical axis shows mel frequencies, and color represents the strength of any given “point” (a given frequency bin at a given point in time).

Note that vowels, being almost harmonic, are represented as a set of ``strips" (the biggest harmonic components, called formants), which both characterise each vowel and are unique for each speaker. Instead, unvoiced consonants, like “p” as in “pet”, are shown like noise, appearing spread among all the frequency bins. Finally, voiced consonants, like ``m" as in “man”, are a mix of harmonics and noise.

The acoustic models for Mel-spectrogram prediction are commonly built using a Seq2Seq encoder-decoder architecture (e.g. Tacotron [3] [4], DeepVoice [5], FastSpeech [6]). The encoder projects the sequence of graphemes or phonemes into a sequence of hidden vectors. Instead the decoder, either autoregressively or in parallel, generates the Mel-spectrogram. Usually, the alignment between encoded graphemes or phonemes and the Mel-spectrogram is done through an additional attention mechanism [7] working between the encoder and the decoder. Additionally, some of these architectures have a separate module to predict the stopping point of the generation process.

In particular, we are interested in the Tacotron 2 architecture [4], as this architecture has proven to be widely extensible [8] and re-usable [9]. Thus, we decided to enhance it with an emotion control module to make the generated voice more expressive.

To complete the speech synthesis pipeline, a vocoder is necessary. Neural vocoders have become a fundamental module of TTS models; they are necessary to synthesise a speech as

[2] A grapheme is “the smallest meaningful contrastive unit in a writing system”. In other words, it is a written symbol that represents a sound; it could be represented by a single letter or a sequence of letters, such “sh”.
[3] A phoneme is any of the perceptually distinct units of sound in a specified language that distinguish one word from another, for example “p”, “b”, “d”, and “t” in the English words “pad”, “pat”, “bad”, and “bat”.

clear as possible [1]. Compared to the previous approach, the Griffin-Lim algorithm [10], neural vocoders yield audio with fewer artefacts and higher quality. Available implementations are based on (dilated) convolutional neural networks (e.g. WaveNet [11], WaveGlow [12], MelGAN [13] [14]) or recurrent neural networks (e.g. WaveRNN [15]). Moreover, they can be trained to work directly on raw waveforms [11] or on Mel-spectrograms [14]. For this work, we leveraged pre-trained implementations of WaveNet and WaveGlow (more on this in Section 4).

2.2. Controlled speech synthesis

Besides the obvious control on the synthesised speech given by the input text (what to say), there are multiple research lines focused on controlling further aspects of the synthesised speech through additional information. These additional aspects can be categorised into two groups: speaker (or timbre, i.e. who is speaking) and prosody (or style, i.e., how to speak) [2]. These two aspects are completely orthogonal and can be combined [8].

Speaker control allows disentangling the content from the speaker, making the overall TTS model more re-usable. The idea is to provide additional information on the speaker to the TTS. This approach allows leveraging multi-speaker data sets, while, previously, neural TTS used to be trained on single-speaker data sets [16]. To implement this kind of conditioning, it is sufficient to concatenate the hidden features extracted from a speaker recognition network to the hidden representation of the input text, as it was done for Tacotron 2 [17].

Prosodic control covers all the aspects concerning intonation, stress, and rhythm, which characterise how a sentence is uttered. These aspects also influence the emotion perceived by the listener. A reference step towards this kind of control on the speaking style is represented by the Global Style Token (GST) [16]. Instead of explicitly modelling the aspect characterising the prosody, this model uses unsupervised style representations learnt alongside the synthesis model. In the original implementation, a Tacotron 2 model was extended with a separate encoder that extracted a vector representing the so-called style embedding; this vector is concatenated to the hidden representation of the input text to provide the decoder with the style information [16]. Notice that this kind of control works at a high-level: the specific changes connected to a given style are learnt and implemented by the spectrogram generator.

In this work, we will focus on prosodic style control. As premised, we are interested in controlling low-level aspects concerning the style of speech by selecting at a high-level the target emotion to express.

3. EMOTRON model

In this section we depict the architectural details of the EMOTRON model for controlled speech synthesis. We describe the architecture of the spectrogram predictor and the architecture of the emotional capturer.

3.1. TTS

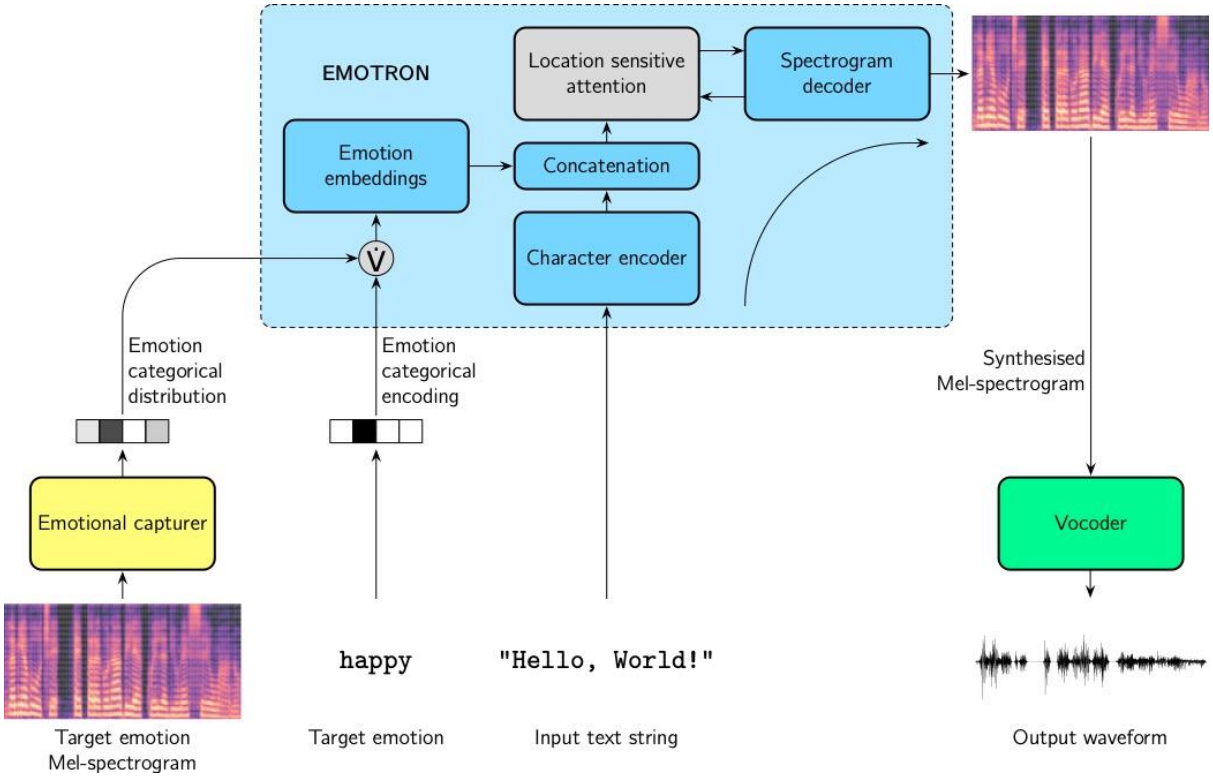

Figure 3.1. EMOTRON high-level architecture and data flow.

We based the EMOTRON architecture on that of Tacoron 2 [4]. The overall network is the same: an encoder-decoder architecture with location-sensitive attention. Similarly to the GST variant [16], we introduced an additional linear transformation to manage the additional emotion input. The overall architecture is depicted in Figure 3.1 We re-used all the hyperparameters from the original implementation.

The input stream of characters $\boldsymbol{x} = (x_1, \dots, x_m)$ representing the text to utter is passed through the Character encoder: a stack of three convolutional layers and a final BiLSTM layer [18]. At this step, the output is a sequence of feature vectors.

We concatenated each hidden vector of the input character stream with the embedding of the target emotion. We extracted such embeddings through a linear transformation taking as input the categorical probability distribution of the desired emotion (predicted by the emotional capturer on a reference Mel-spectrogram; more on this in Section 3.2) $\boldsymbol{e} = (e_1, \dots, e_k)$, where $\boldsymbol{e} \in [0,1]^k \subseteq \mathbb{R}^k : \sum_{i=1}^{k} e_i = 1$. In this way, we can tell the network to imitate the emotion found in a reference audio clip. In particular, we considered $k = 4$ different emotions (namely "neutral", "sadness", "anger", and "happiness") in our implementation, and generated 32-dimensional embeddings. Note that, alternatively, such distribution can actually be a one-hot encoding of the target emotion ($\boldsymbol{e} \in \{0,1\}^k$). In this way, we cal tell the network to generate a specified emotion.

The hidden vector generated above is the input that guides the decoding of the output Mel-spectrogram. The alignment between the encoder and the decoder is realised through the location sensitive attention [4]. All the hyperparameters at this step were left unchanged.

The Spectrogram decoder is designed to work with a causal approach: it leverages all the Mel-spectrogram $\boldsymbol{Y}_{t'<t}$ up to the current step t in the sequence to predict the next slice $\hat{\boldsymbol{y}}_t$. The input Mel-spectrogram up to the latest generated step $\boldsymbol{Y}_{t'<t}$ is passed through a stack of two pre-net layers, two LSTM [19] layers (aligned with the attention) and a final linear projection. The final output is further refined through a stack of five convolutional post-net layers. An additional linear projection on top of the latest LSTM output is used as a stop-net to signal the end of the generation process. All the hyperparameters at this step were left unchanged.

3.2. Emotional capturer

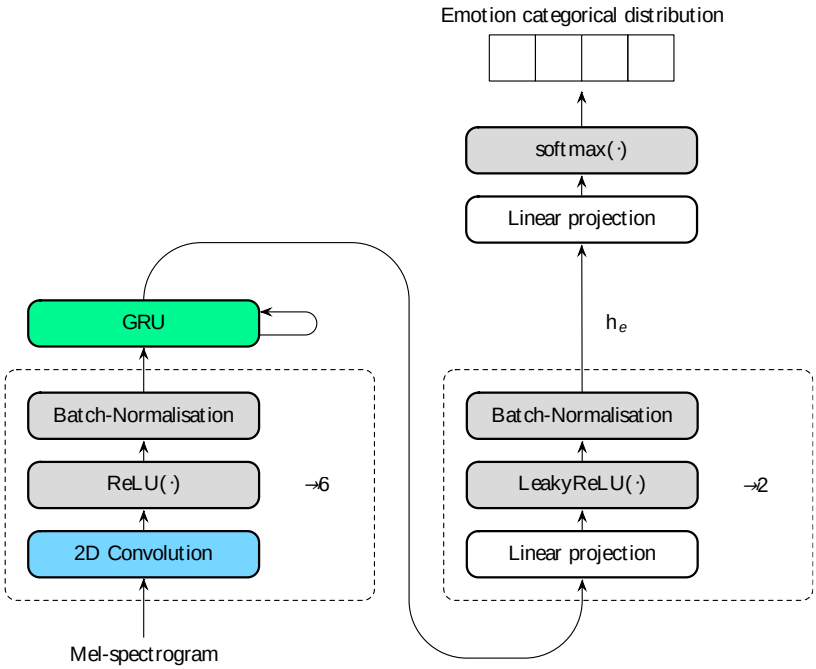

Figure 3.2. Architecture of the emotional capturer.

The emotional capturer plays two essential roles in the EMOTRON architecture.

On one side, it takes care of extracting the perceived emotion probability distribution $\boldsymbol{e} \in [0,1]^k \subseteq \mathbb{R}^k : \sum_{i=1}^{k} e_i = 1$ from the Mel-spectrogram $\boldsymbol{Y} \in \mathbb{R}^{C \times T}$ of a reference audio clip (where C is the number of frequency bins in the Mel-spectrogram and T is the length of the clip), so it is a discriminative neural network for emotion recognition. Thus, it allows replicating the perceived emotion from the reference clip during the synthesis process. On the other hand, this network's hidden features $\boldsymbol{h}_e$ can be leveraged to compute the style loss. As we explain in Section 4, this loss is fundamental to enforcing the emotion conditioning when training the whole network.

The emotion capturer network comprises a stack of six 2D convolutional layers that takes the Mel-spectrogram as input. A GRU layer and four fully connected layers complete the sequence of transformations. For further details about the architecture, refer to Figure 3.1.

The number of output channels in the convolutional layers doubles every two convolutional blocks (the dashed blue box in Figure 3.2), starting from 32. Each 2D convolution uses a 3×3 kernel and a 2×2 stride. After every convolution we apply ReLU(·) activation and batch normalisation.

The GRU layer generates a sequence composed of 128-dimensional vectors (one for each time slice); we take only the last one to summarise the entire sequence.

Of the following two linear blocks (the dashed orange box), the first one has, again, 128-dimensional vectors, while the last one yields 256-dimensional vectors. This feature vector represents $\boldsymbol{h}_e = h_e(\boldsymbol{Y}) \in \mathbb{R}^{256}$: the hidden representation of the whole sequence. Note that such linear blocks use a LeakyReLU(·) activation and dropout regularisation.

Finally, a following liner projection yields the logits of the four considered emotions, and a softmax(·) activation allows to output their posterior probabilities $\boldsymbol{e}$.

The overall architecture is agnostic of the actual labels it learns to discriminate. Conceptually, the emotion capturer acts as the GST module from Tacotron 2 [16]. However, GST learns such labels through an unsupervised approach while training the entire model; the emotion capturer, instead, can be instructed on the target labels, independently of what they represent.

We also trained a second model using a semi-supervised learning approach, for the Tacotron 2 baseline model; we created alternative labels by clustering the clips in the data set using common emotion discriminative features (more on this in Section 5). We used this pseudo-emotional capturer to add emotion control to the Tacotron 2 baseline model.

4. Training and inference

This section provides the details concerning the models' training (both EMOTRON and the emotional capturer) and inference (i.e., audio synthesis).

4.1 Training details

We designed the EMOTRON model to enforce emotion control on synthesised audio, as premised. We resorted to results coming from computer vision to achieve this goal [20]. We approached our problem following insights comings from image style transfer models: we trained our network to minimise, at the same time, a content loss $\mathcal{L}_{content}(\cdot)$ and a style loss $\mathcal{L}_{style}(\cdot)$, as described in the following equation; where $\boldsymbol{x}$ is the input sequence of characters, $\boldsymbol{e}$ is the target emotion categorical distribution, $\boldsymbol{Y}$ is the reference Mel-spectrogram, and $\widehat{\boldsymbol{Y}} = f_{TTS}(\boldsymbol{x}, \boldsymbol{e}|\boldsymbol{Y})$ is the output of EMOTRON, generated with guided decoding (more on this later).

$$\mathcal{L}_{TTS}(\boldsymbol{x}, \boldsymbol{Y}, \boldsymbol{e}) = \mathcal{L}_{content}(\boldsymbol{Y}, f_{TTS}(\boldsymbol{x}, \boldsymbol{e}|\boldsymbol{Y})) + \mathcal{L}_{style}(\boldsymbol{Y}, f_{TTS}(\boldsymbol{x}, \boldsymbol{e}|\boldsymbol{Y}))$$

The content loss is the L_2 norm of the reconstruction error between target and predicted spectrograms, $\boldsymbol{Y}$ and $\widehat{\boldsymbol{Y}}$. The objective of this loss is to train the spectrogram predictor to output intelligible audio, and it is the usual loss used to train neural TTS models. We reported the formulation in the following equation, where C is the number of frequency bins forming the Mel-spectrograms (predefined and fixed), T is the number of time slices for the current reference Mel-spectrogram (different for each spectrogram we consider), while $y_{c,t}$ and $\hat{y}_{c,t}$ are the value of the c-th frequency bin of the t-th time slice, obtained from the reference Mel-spectrogram and generated by our model, respectively.

$$\mathcal{L}_{content}(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}) = \left\|\boldsymbol{Y} - \widehat{\boldsymbol{Y}}\right\|_2 = \sqrt{\sum_{t=1}^{T}\sum_{c=1}^{C}\left(y_{c,t} - \hat{y}_{c,t}\right)^2}$$

At training time, the network uses guided decoding to generate $\widehat{\boldsymbol{Y}}$. The Spectrogram decoder inside the EMOTRON model is designed to generate in an autoregressive manner, consuming as internal input the last generated Mel-spectrogram time slice. During training, however, we use the reference Mel-spectrogram time slice instead of recurring the output slice.

Instead, we used the style loss to enforce the emotion based (or emotion related, in the case of the pseudo-emotional capturer) style control on prosody. The idea behind this loss is to find space where it is possible to capture emotional information, like the hidden representation of a speech emotion discriminator. If the projection in such a space is a differentiable transformation (as in our case), it is possible to compute the style loss from the different hidden representations. We follow the approaches proposed for image style transfer [21] [20]; we compute the style loss as the L_2 norm of the difference between the Gram matrices $\boldsymbol{G}(\cdot) \in \mathbb{R}^{256\times256}$ obtained by the two hidden vectors of the reference and generated Mel-spectrograms. We reported the formulation of this loss in the following equation.

$$\mathcal{L}_{style}(\boldsymbol{Y}, \widehat{\boldsymbol{Y}}) = \left\| \boldsymbol{G}(h_e(\boldsymbol{Y})) - \boldsymbol{G}(h_e(\widehat{\boldsymbol{Y}})) \right\|_2 = \sqrt{\sum_{i=1}^{256}\sum_{j=1}^{256}\left(g_{i,j} - \hat{g}_{i,j}\right)^2}$$

We trained the EMOTRON network for almost 120k update steps on mini-batches of 32 audio clips. We set the learning rate to $\eta = 10^{-3}$, with an exponential decay to 10^{-6}. To enforce regularisation we used weight decay with $\lambda = 10^{-6}$ and we clipped the maximum norm of the gradients to 1.

Such training procedure only affects the EMOTRON network $f_{TTS}(\cdot)$ (i.e., the encoder-decoder) and the emotion embeddings.

Instead, the emotional (and pseudo emotional) capturer $f_e(\cdot)$ was trained separately to discriminate among the considered emotions (or emotion relate clusters). We applied the usual negative log-likelihood loss computed through the cross-entropy on the target class. The following equation describes the loss, where e_i (with $i \in [1, k] \subseteq \mathbb{N}$, and $k = 4$ for the emotional capturer or $k = 3$ for the pseudo emotional capturer) is the i-th emotion associated with the speech signal represented through $\boldsymbol{Y}$, while $f_e(\cdot)_i$ is the reconstructed probability for such i-th emotion.

$$\mathcal{L}_e(e_i, \boldsymbol{Y}) = -\ln P(e_i|\boldsymbol{Y}) = -\ln f_e(\boldsymbol{Y}_i)$$

4.2. Inference details

During inference, the model leverages auto-regressive decoding (i.e., it recurs the latest output time slice of the Mel-spectrogram as the next input of the decoder). Thus, instead of generating with a guided approach as in training and computing $\widehat{\boldsymbol{y}} = f_{TTS}(\boldsymbol{x}, \boldsymbol{e}|\boldsymbol{Y}_{t'<t})$, it computes $\widehat{\boldsymbol{y}} = f_{TTS}(\boldsymbol{x}, \boldsymbol{e}|\widehat{\boldsymbol{Y}}_{t'<t})$ to generate the Mel-spectrogram $\widehat{\boldsymbol{Y}}$. The autoregressive process continues until the stop-net triggers the interruption (predicting a sufficiently high posterior "stop" probability).

Concerning emotion control, since we followed the same approach of GST, it is possible to obtain it in two ways. The former approach consists in providing a reference audio clip spectrogram to replicate its emotional style. The emotional capturer takes care of extracting the high-level information necessary to feed the model. This approach is followed during training

and can be used for inference. Alternatively, the latter approach prescribes feeding the model with the categorical emotion and fetching the corresponding embedding to be concatenated to the textual features to condition the synthesis. We followed this approach only during inference.

Since EMOTRON outputs the Mel-spectrogram, to extract the raw audio waveform, generating the final synthesised speech, we leveraged a vocoder. During testing and inference, we leveraged the WaveNet vocoder [11] because of the higher audio quality (during training, to get samples faster, we employed WaveGlow [12]). Notice that we did not tie the model to a specific vocoder, and thus this final module can be freely substituted.

5. Data

In this Section, we present the corpus we employed to train the EMOTRON TTS model and the connected capturers. We organised the sections according to the target model.

5.1. Conditioned Speech synthesis

The data set we selected to train EMOTRON is the one released for the Blizzard Challenge 2013 [22]. It is a collection of audiobooks read by a single speaker, with high expressivity. The data set included more than 100 hours of recorded clips.

We selected this data set for its size and high expressivity of uttered sentences. The high number of samples and their variety is crucial to have the network properly learning the different styles. Moreover, since we are not modelling multi-speaker properties, having a single speaker data set is crucial for convergence.

We retrieved two subsets of the original corpus: the "selected" version and the "full" version. The former is a selection of clips already preprocessed and paired with the transcription. The latter required some preprocessing to be usable, because the clips contained the reading of entire chapters; in particular, we did:

- transcripts retrieval from the Project Gutenberg website[4];
- transcripts alignment using the Aeneas forced aligner[5];
- clips cropping to have easy to process small utterances;
- post-processing to filter out clips shorter than 1 s and longer than 14 s.

The overall data set resulted in 120 hours of recordings.

5.2. Emotion recognition for EMOTRON

We required emotion labels on the speech synthesis data set to train the emotional capturer. Since these labels were not available and due to the size of the data set, we resorted to automatic systems. We leveraged two neural networks for emotion recognition from speech (and text): PATHOSnet [23] and CNN-MFCC [24]. We used these two systems for a more robust result.

PATHOSnet distinguishes among "neutral", "happiness", "sadness" and "anger", while CNN-MFCC can identify also "calm", "fear", "disgust" and "surprise". To have unified labels, we combined "calm" and "neutral", and "sad" and "fearful". Additionally, we removed "surprise" due to its low presence in the data set.

[4] http://gutenberg.org/
[5] https://www.readbeyond.it/aeneas/

We used these two networks to label the audio clips in the corpus by combining their predictions. Given the individual reported results, we applied the following rules to combine the predictions:

- If the predicted label is not “happy” use the prediction from PATHOSnet;
- else use the prediction from CNN-MFCC.

5.3. Emotion-related clusters recognition for the baseline model

The original GST approach for expressive speech leveraged unsupervised learning to identify the style labels while training the speech synthesis model. Here instead, we propose splitting this step and lean pseudo-emotion labels.

Instead of relying on the hidden features learnt by the deep style model, we performed clustering on the audio clips and learnt to predict the cluster labels. We extracted features that correlate with emotion from the audio clips applied feature reduction and clustered clips on these representations.

We used the OpenSmile tool [25] to extract all the 120 features including intensity, loudness, MFCCs, pitch and pitch envelope, probability of voicing, pitch, line spectral frequencies, zero-crossing rate. We applied PCA to reduce the features to 10 (retaining 90% of variance), and we used k-Means clustering with $k = 3$ (we also experimented with $k = 2$ and $k = 4$). Finally, we post-processed the result by deleting the clips too close to the boundaries between clusters.

6. Evaluation approach

This section outlines the approach we followed to assess the quality of the synthesised speech and the clarity of the emotion conditioning. We evaluated through a survey on a website we created specifically for this evaluation. The survey was composed of 12 questions, three for each considered emotion.

The website displayed a button to play the audio clip associated with the question and displayed the corresponding transcription of the clip. Under each clip, human listeners could provide their opinion scores about speech quality and recognised emotion. We aggregated the evaluations into Mean Opinion Scores (MOS) we reported and commented on in Section. 7.

To assess the quality of our model, we compared it against ground truth --real clips uttered by humans taken from the Blizzard Challenge 2013 corpus-- and a baseline TTS model. As a baseline expressive TTS, we opted for Tacotron 2 with pseudo-emotion labels. This system uses the clusters extracted from the set of reference clips as target labels to add expressivity to the synthesised speech, conditioning the decoding on the identified style. We selected this model because it is based on Tacotron 2 and because it leverages a mechanism close to that of GST, which is considered state-of-the art in terms of expressive TTS (the difference is that our approach applies the unsupervised step of clustering separately from the speech synthesis training). Additionally, we considered it also because it leverages the same core architecture of EMOTRON, enabling a direct comparison between the pseudo emotion clusters and actual emotion labels used for conditioning.

To achieve the best results from EMOTRON, we leveraged the WaveNet vocoder to convert the Mel-spectrogram into a waveform. For better comparison, we used the same vocoder for the baseline TTS with pseudo labels.

6.1. Speech quality

To assess the quality of the EMOTRON spectrogram predictor, we used the human listener's MOS. We asked each listener to rate the quality of the audio clip on a 1-to-5 scale with a unit increment. The idea is to indirectly evaluate the quality of the spectrogram predictor by evaluating the audio quality.

To help listeners in the evaluation, we provided the following explanation for the scoring system:

1. **bad**, unrecognisable speech;

2. **poor**, speech is barely recognisable;
3. **fair**, acceptable quality of speech, small errors allowed;
4. **good**, speech with proper pronunciation and quality;
5. **excellent**, perfect speech, sounds natural and expressive.

6.2. Emotional clarity

To assess the quality of the EMOTRON emotion conditioning modules and the overall emotional clarity of the synthesised speech, we calculated the human listener's average emotion recognition accuracy. We asked the listeners to associate one of the four possible emotion labels to each clip: neutral, sad, angry and happy.

We synthesised EMOTRON's clips conditioning it on the target emotion expressed in a reference clip from the ground truth. The baseline TTS was conditioned on the emotional labels detected by the pseudo-emotional capturer. We used the same clips of the speech quality evaluation, including the ground truth.

7. Results

We collected responses from 54 listeners during the three days the survey website was online. In the following, we report and comment on the final scores.

7.1. Speech quality

Table 1 - Results of human evaluation on speech quality

	Speech quality: MOS				
	Neutral	**Sadness**	**Anger**	**Happiness**	**Average score**
Ground truth	4.50	4.75	4.20	4.60	4.53
Tacotron 2 w/ pseudo-emo.	4.12	3.89	4.08	3.77	3.96
EMOTRON	4.10	4.00	3.71	3.65	3.87

We reported the MOS collected during the evaluation in Table 1. We reported the quality scores averaged emotion-wise and averaged on the entire support.

Ground truth consistently outperforms both TTS models. This result is not surprising considering that these clips are actual human voice samples. Ground truth relative performances are 14.4% and 17.1% better than Tacotron 2 with GST and EMOTRON, respectively. Looking at the emotion-wise breakdown, we can see that the highest score was on "sadness" clips and the worst score on "anger" clips; yet, any single ground truth score is higher than any of the two TTS models.

EMOTRON underperforms the baseline TTS in the audio quality evaluation. However, the relative performance difference with respect to this baseline TTS is only -2.3%, meaning that the two systems deliver audio clips with very similar quality. Moreover, since the spectrogram predictor is the same between the two models. This is probably due to the fact that the style conditioning system we developed affects audio quality independently of the target style.

7.2. Emotional clarity

We reported the accuracy scores collected during the evaluation in Table 2. We reported the clarity scores averaged emotion-wise and averaged on the entire support.

Table 2 - Results of automatic emotion evaluation on emotion clarity.

	Emotion clarity: recognition accuracy [%]

	Neutral	Sadness	Anger	Happiness	Average score
Ground truth	89	78	63	80	78
Tacotron 2 w/ pseudo-emo.	54	48	49	41	48
EMOTRON	70	51	55	46	56

We leverage the listener's emotion recognition accuracy as a scoring function, as premised. If we consider the results on ground truth, accuracy scores are similar to those of other works, where humans reported a recognition accuracy of ~70.0% [26].

As happened for speech quality, ground truth outperforms both TTS models in emotional clarity. Ground truth relative performances are 62.5% and 39.3% better than the baseline TTS and EMOTRON, respectively. Looking at the emotion-wise breakdown, we can see that the highest score was on "neutral" clips for all systems. Instead, the worst score is on "anger" clips for ground truth, while both TTS models perform worse on "happiness" clips.

Unlike speech quality, EMOTRON outperforms the baseline TTS in the audio quality evaluation, and it does so by a consistent margin. The relative performance difference with respect to this baseline TTS is 16.7%. Since the two TTS models share the same spectrogram predictor architecture and were trained on similar data sets, we can hypothesis that the unsupervised model learnt by clustering the clips did not reflect the division among different emotions despite the selected features. To fully understand the reason behind this difference, an ablation study on the two networks would be necessary.

8. Conclusion

In this paper, we presented EMOTRON: a TTS with emotion conditioning. EMOTRON builds on top of a well-known neural TTS architecture (Tacotron 2) to synthesise expressive speech. We extended the base architecture to enhance the expressivity of the uttered text by training it to synthesise audio given an input text and the emotion to display.

During training, we used a combination of Mel-spectrogram reconstruction and style losses; in this way, we enforced generative capabilities and emotion control on EMOTRON. We used the Mel-spectrogram reconstruction loss to have the model learn generative capabilities, together with the style loss to measure the distance between the enforced emotion and the desired one. We computed it through a separate neural network designed to discriminate among emotions from speech. We used the style loss to have the model learn how add expressivity in the synthesised speech.

To assess the quality of our model, we resorted to human evaluation to measure speech quality and emotional clarity. In the evaluation we compared EMOTRON with natural human speech and another TTS for expressive speech built on the same architecture leveraged by EMOTRON. The baseline system uses pseudo-emotional labels learnt through clustering.

Both TTS models performed worse than human speech in terms of synthesised audio quality and emotional clarity, thus leaving space for many improvements. Instead, the two TTS performed comparably in speech quality (despite the baseline TTS being slightly better). EMOTRON, however, outperformed the baseline TTS by a consistent margin in terms of emotional clarity. This result highlighted that the use of unsupervised labels (as we did for the pseudo emotional capturer), despite being useful in general, provided worse results than an approach based on supervised learning (like the one leveraged by EMOTRON) when such labels represent emotions.

References

[1] D. Jurafsky e J. H. Martin, «Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (3rd Edition),» 2022.

[2] X. Tan, T. Qin, F. K. Soong e T.-Y. Liu, «A Survey on Neural Speech Synthesis,» *CoRR*, vol. abs/2106.15561, 2021.

[3] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark e R. A. Saurous, «Tacotron: Towards End-to-End Speech Synthesis,» in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J.-S. Ryan, R. A. Saurous, Y. Agiomyrgiannakis e Y. Wu, «Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions,» in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 2018.

[5] W. Ping, K. Peng, A. Gibiansky, S. Ö. Arik, A. Kannan, S. Narang, J. Raiman e J. Miller, «Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning,» in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[6] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao e T.-Y. Liu, «FastSpeech: Fast, Robust and Controllable Text to Speech,» in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[7] D. Bahdanau, K. Cho e Y. Bengio, «Neural Machine Translation by Jointly Learning to Align and Translate,» in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark e R. A. Saurous, «Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron,» in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018.

[9] A. Favaro, L. Sbattella, R. Tedesco e V. Scotti, «ITAcotron 2: Transfering English Speech Synthesis Architectures and Speech Features to Italian,» in *4th International Conference on Natural Language and Speech Processing, Trento, Italy, November 12-13, 2021*, 2021.

[10] D. W. Griffin e J. S. Lim, «Signal estimation from modified short-time Fourier transform,» in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '83, Boston, Massachusetts, USA, April 14-16, 1983*, 1983.

[11] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior e K. Kavukcuoglu, «WaveNet: A Generative Model for Raw Audio,» in *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, 2016.

[12] R. Prenger, R. Valle e B. Catanzaro, «Waveglow: A Flow-based Generative Network for Speech Synthesis,» in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, 2019.

[13] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio e A. C. Courville, «MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis,» in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[14] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen e L. Xie, «Multi-Band Melgan: Faster Waveform Generation For High-Quality Text-To-Speech,» in *IEEE Spoken Language Technology Workshop, SLT 2021, Shenzhen, China, January 19-22, 2021*, 2021.

[15] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman e K. Kavukcuoglu, «Efficient Neural Audio Synthesis,» in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018.

[16] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren e R. A. Saurous, «Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis,» in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018.

[17] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Lopez-Moreno e Y. Wu, «Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis,» in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018.

[18] M. Schuster e K. K. Paliwal, «Bidirectional recurrent neural networks,» *IEEE Trans. Signal Process.*, vol. 45, p. 2673–2681, 1997.

[19] S. Hochreiter e J. Schmidhuber, «Long Short-Term Memory,» *Neural Comput.*, vol. 9, p. 1735–1780, 1997.

[20] L. A. Gatys, A. S. Ecker e M. Bethge, «Image Style Transfer Using Convolutional Neural Networks,» in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

[21] J. Johnson, A. Alahi e L. Fei-Fei, «Perceptual Losses for Real-Time Style Transfer and Super-Resolution,» in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, 2016.

[22] S. King e V. Karaiskos, «The Blizzard Challenge 2013,» *festvox*, 2013.

[23] V. Scotti, F. Galati, L. Sbattella e R. Tedesco, «Combining Deep and Unsupervised Features for Multilingual Speech Emotion Recognition,» in *Pattern Recognition. ICPR International Workshops and Challenges - Virtual Event, January 10-15, 2021, Proceedings, Part II*, 2020.

[24] M. G. de Pinto, M. Polignano, P. Lops e G. Semeraro, «Emotions Understanding Model from Spoken Language using Deep Neural Networks and Mel-Frequency Cepstral Coefficients,» in

2020 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2020, Bari, Italy, May 27-29, 2020, 2020.

[25] F. Eyben, F. Weninger, F. Gross e B. Schuller, «Recent Developments in OpenSMILE, the Munich Open-Source Multimedia Feature Extractor,» in *Proceedings of the 21st ACM International Conference on Multimedia*, New York, NY, USA, 2013.

[26] V. Chernykh, G. Sterling e P. Prihodko, «Emotion Recognition From Speech With Recurrent Neural Networks,» *CoRR*, vol. abs/1701.08071, 2017.