Real-Time Voice Cloning for Spanish Speakers

Alex Steve Chung Alvarez, Manolo Canales Cuba

December, 2021

Thesis Seminar II

Abstract

Text to Speech has been a very popular field for research in the last three years. However, there are not many text to speech projects focused in the spanish language. The main problem of text to speech is that a big dataset of a target voice is needed in order to train a model with that voice. To overcome this problem, real-time voice cloning has been studied since 2018. In that year, the first implementation of real-time voice cloning was published. We use this open source code to train a spanish model of the synthesizer for real-time voice cloning. We provide the first pre-trained spanish model of the synthesizer to the community.

Contents

List of Figures

1 Introduction

Voice Cloning to develop Text-to-Speech (TTS) systems is a very popular field in research since 2018 ([10],[14],[15],[17],[20],[13]). Most of the studies use deep learning pipelines to achieve voice cloning, though, a recent study has been done with meta-learning [13]. In this field, deep learning is needed to achieve a better voice quality, to produce natural sounding voices and to reduce inference time. The most popular datasets for study are LibriSpeech [19], VCTK [25], VoxCeleb [16], VoxCeleb2 [6], LibriTTS [26], LJSpeech [9], among others. All of which are datasets in English language, then most of the projects are focused on this language. There are also some projects published for Chinese language such as [22], but there are no publications made for the Spanish language. For this reason, we searched for datasets in Spanish such as tux100h (`https://discourse.mozilla.org/t/sharing-my-100h-of-single-speaker-spanish/45288`), commonvoice (`https://commonvoice.mozilla.org/es/datasets`) and Crowdsourcing Peruvian Spanish for Low-Resource Text-to-Speech [8].

One big problem of generating text to speech models is that long hours of speech are required for training them. Even in voice cloning, this is still an issue. The most common solution to this issue is to fine-tune the model with hours of speech of the target voice. This is used by many frameworks already such as AdaSpeech [3], FastSpeech2 [4], MozillaTTS (`https://github.com/mozilla/TTS`), Coqui TTS (`https://github.com/coqui-ai/TTS`).

Our goal is to achieve voice cloning of a Spanish speaker with just an audio of a few seconds length. In order to achieve results comparable to voice cloning of an English speaker, we use the Real-Time Voice Cloning (RTVC) repository (`https://github.com/CorentinJ/Real-Time-Voice-Cloning`) which is explained in [10] and is an implementation of [11]. This study shows a three-stage pipeline that allows voice cloning of an unseen speaker during training from only a few seconds of speech and without the need of retraining the model. These three stages are: the encoder, the synthesizer and the vocoder. We train only the synthesizer in order to let the model produce spanish speaking from text.

The structure of this document goes as follows. We begin with some definitions of theoretical concepts needed to understand the project. Follows a brief explanation of the pipeline. We then present our results of attention and the generated mel-spectrogram. We conclude with a presentation of the toolbox implemented by Jemine demonstrating that it works in spanish, even when the encoder is not trained with a spanish dataset.

2 Review of Theoric Definitions

2.1 Speaker Embeddings

Embeddings are meaningful representations of the voice of a speaker [10]. They are a well-stablished approach to encode discriminative information in speakers and let the model be speaker independent [1]. These are vectors that will be used to represent the different speakers in the latent space and to condition the generated spectrograms to the target voice.

2.2 Waveforms

Waveforms are the typical representations of audios in terms of Amplitude along time. Audio happens to be difficult to model since amplitude vs time is a particularly dense domain and audio signals are typically highly nonlinear [10]. Then, spectrograms are most commonly used instead of waveforms. As an example of waveforms, see the first image of Figure 1.

2.3 Spectrograms

Spectrograms are representations of the audios in terms of frequency, time and amplitude. The most common spectrograms used in these researches ([10],[14],[15],[17],[20],[13]) are Mel-Spectrograms, which are spectrograms in a logarithmic/Mel (melody) scale. A representation that brings out features in a more tractable manner is the time-frequency domain. Then, spectrograms are smoother and much less dense than their waveform counterpart [10]. As an example of spectrograms, see the last image of Figure 1.

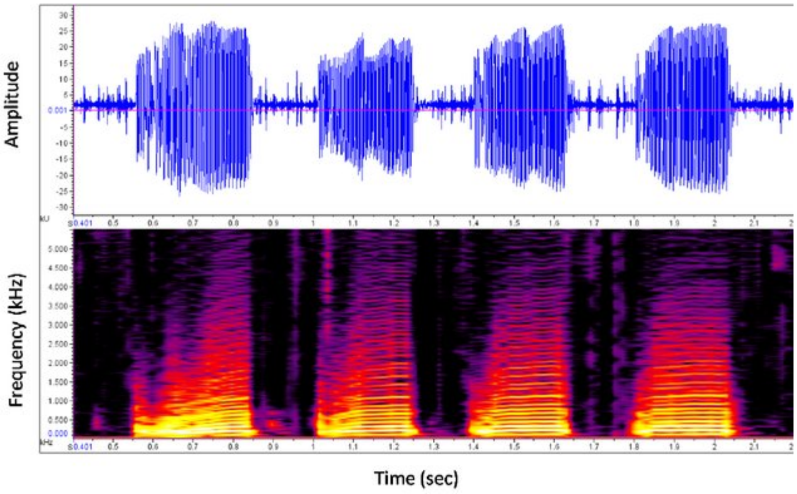


Figure 1: A waveform and a spectrogram representing the same sound.
Source: Figure 2 [12]

2.4 The attention alignment graph

The attention alignment graph explains which letters of the input text would be synthesized by the model at each point in time.
When you see the familiar line, it means the model has learned attention. That means the model knows how long to synthesize each character of input text, and when to move on. It is necessary for synthesizing new texts unseen in training. Attention was introduced by Bahdanau et al. as a mechanism meant to link decoder outputs to the encoder outputs. Figure 2 is an example of attention learned by a model.

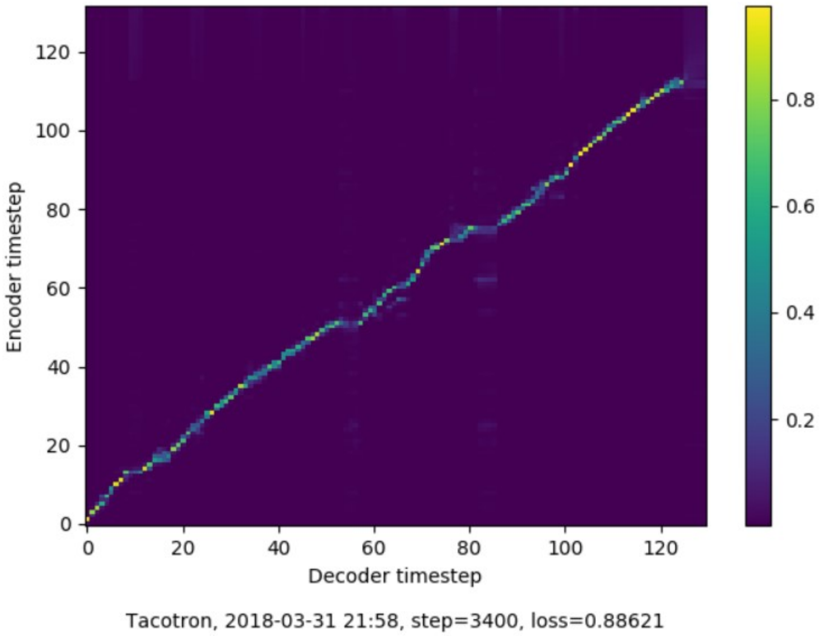


Figure 2: Example of Attention learned by a model.
Source:`https://github.com/Rayhane-mamah/Tacotron-2/wiki/Spectrogram-Feature-prediction-network`

3 Pipeline

The framework is composed of three stages: an encoder (described in [23]), a synthesizer (described in [21]) and a vocoder (described in [18]), each of these stages can be trained independently from the others. Figure 3 is a visual representation of the model. A more detailed explanation of the pipeline can be found in [10].

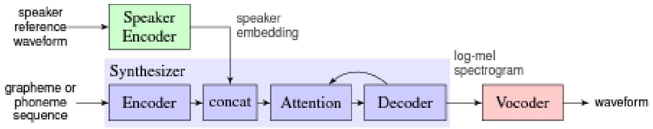


Figure 3: General vision of the model. Each module is trained independently from the other.
Source: Figure 1 [11]

3.1 Encoder

The encoder is the first stage of the pipeline, in this stage the target voice is identified by using the Generalized End-to-End loss function [23]. It needs to be capable of producing an embedding from a short utterance of the target speaker, also must be robust to noise. Then it derives this embedding to the synthesizer. The model for the encoder takes the most of the time to train since it needs more data in order to differenciate well multiple speakers. For this reason, we used the pretrained model of Jemine, which was trained with LibriSpeech-Other [19], VoxCeleb1 [16] and VoxCeleb2 [6]. For further explanation of the encoder, refer to [10].

3.2 Synthesizer

The next step of the pipeline is the synthesizer. Here the embedding from the target voice is used to condition the generation of the mel-spectrogram. The synthesizer takes the required text and asociate them to the phonemes learned in training, then it produces a mel-spectrogram which will be passed to the vocoder, this mel-spectrogram is conditioned by the embedding from the target voice. The datasets used to train the synthesizer need to have good quality, since it will define the quality of the output, also need to have a large variety of speakers, so it can produce a more similar mel-spectrogram to the target speaker embedding and hence generate a similar voice. To train the synthesizer with different datasets than the ones used in [10], each new dataset needs to have the following folder structure:

```
datasets_root
    * new_dataset
        * subfolder
            * subfolder2
                * subfolder3
                    * utterance-001.wav
                    * utterance-001.txt
                    * utterance-002.wav
                    * utterance-002.txt
                    * utterance-003.wav
                    * utterance-003.txt
```

(This structure is from LibriSpeech [19] dataset, which was used for training the synthesizer in [10]).
Once we have the dataset in the correct structure, we need to preprocess the audios (to extract the mel-spectrograms from each of them) and to generate the embeddings for each audio. Once everything is set-up, we can start training the Tacotron model (Figure 4), described in [21] and [24], with the new dataset. To train a new model of the synthesizer takes from three days (if it is a small dataset) to more than one week (for big datasets), it is highly recommended by the community to look for a big dataset with a large variety of speakers and at least 300 hours of audios. With small datasets attention can be achieved, therefore the output will sound natural, but it will be different from the target speaker. Tacotron is used because it usually operates faster than real-time [10].

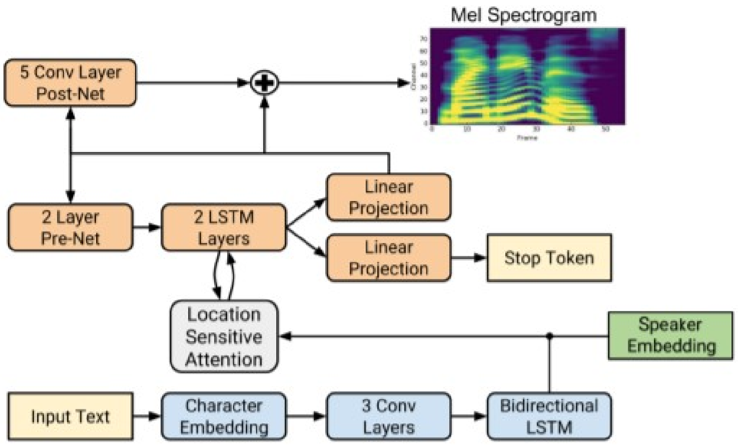


Figure 4: The modified Tacotron architecture. The blue blocks correspond to the encoder and the orange ones to the decoder.
Source: Figure 15 [10] (extracted and modified from [21])

3.3 Vocoder

The vocoder does the oposite of the encoder, it generates audio waveforms from mel-spectrograms. Even though, raw waveforms can be generated by using the Griffin-Lim algorithm [7], it is better to use a deep learning model as a vocoder, so we can get more natural outputs. The model used in [10] is an alternative WaveRNN, which is based on WaveRNN [5]. Refer to [10] for a deeper explanation on how the alternative WaveRNN works. We use, again for the vocoder, the pretrained model of Jemine.

4 Experiments and Results

For training in Spanish, we looked for different datasets, such as tux100h (https://discourse.mozilla.org/t/sharing-my-100h-of-single-speaker-spanish/45288), common-voice (https://commonvoice.mozilla.org/es/datasets) and Crowdsourcing Peruvian Spanish for Low-Resource Text-to-Speech [8]. However, tux100h was composed just by one speaker, so it was discarted for this document (though, it presented good results for a text-to-speech voice in spanish after training 50k steps). The Crowdsourcing Peruvian Spanish for Low-Resource Text-to-Speech was a promising dataset, since it was composed of multiple peruvian spanish speakers, though, it seems that it doesn't have enough speakers, so it produces a clear and natural voice, but it does not sound at all like the target voice, so we stopped the training at 50k steps. The best dataset was the common-voice (train) dataset released by mozilla, since the outputs sound more likely to the target utterances, so we will only discuss the results from this dataset. By the time we were writing this document, a member from the community, @racoonML, shared a pretrained spanish model from the Multilingual LibriSpeech Spanish dataset, he trained the complete pipeline and obtained similar results to ours. To hear the inference results from the models of the common-voice and the peruvian spanish datasets along with the pretrained models shared by Jemine and racoonML, please visit (https://alexstevechungalvarez.github.io/Real-Time-Voice-Cloning-Spanish/).
The repository with the code for preparing the dataset and the source code forked from the original RTVC with modifications in order to work in spanish is https://github.com/AlexSteveChungAlvarez/Real-Time-Voice-Cloning-Spanish.
We trained the synthesizer with the common-voice dataset in spanish (we used the train set, as a sugestion (https://github.com/CorentinJ/Real-Time-Voice-Cloning/issues/941#issuecomment-989855239) from Bebaam (https://github.com/Bebaam), who is an active member of the RTVC community) for over 200k steps with a batch size of 12 on 1 GPU NVIDIA RTX 3060 and 1 GPU NVIDIA RTX 2060. The first 100k steps took about half a week to train, the rest, about a week, and to prepare the dataset it took about half a week. We could train the first 100k steps with the contribution of Andredenise (https://github.com/Andredenise) who is a user of github from the RTVC community that let us use his computational power. He also collaborated in training the 50k steps of the Peruvian Spanish dataset.
To confirm the common-voice dataset had a variety of accents and gender, we made some plots from it. Figures 5 and 6 show that the dataset indeed has a great variety of both accents and gender, though, there was an unknown accent at the moment of doing the plots which seems to be the spanish accent, since most of the results after training sound more likely to a spanish accent and this unknown accent has the most quantity of utterances in the train set. This set has a total of 196006 utterances.

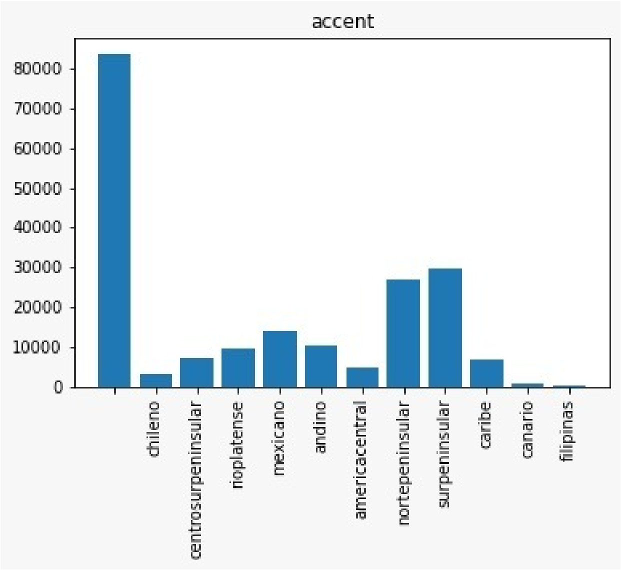


Figure 5: Variety of accents of the Common-Voice dataset train set

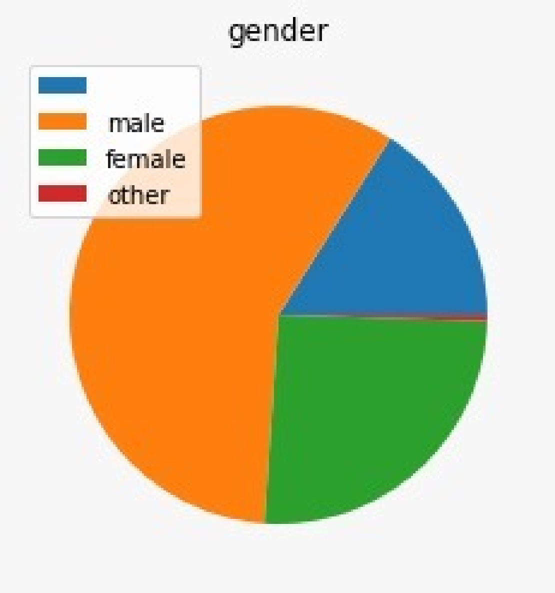


Figure 6: Variety of gender of the Common-Voice dataset train set.

To compare the results with Jemine's, the Mean Opinion Score (MOS) would have been the correct formal evaluation. However, he couldn't calculate it because of lack of time and neither could we, also for lack of time. Then, we will share some of the plots obtained for attention and Mel-Spectrograms during training. Figure 7 shows the attention graph for step 210.5k, while Figure 8 shows the target and generated mel-spectrograms for that step.

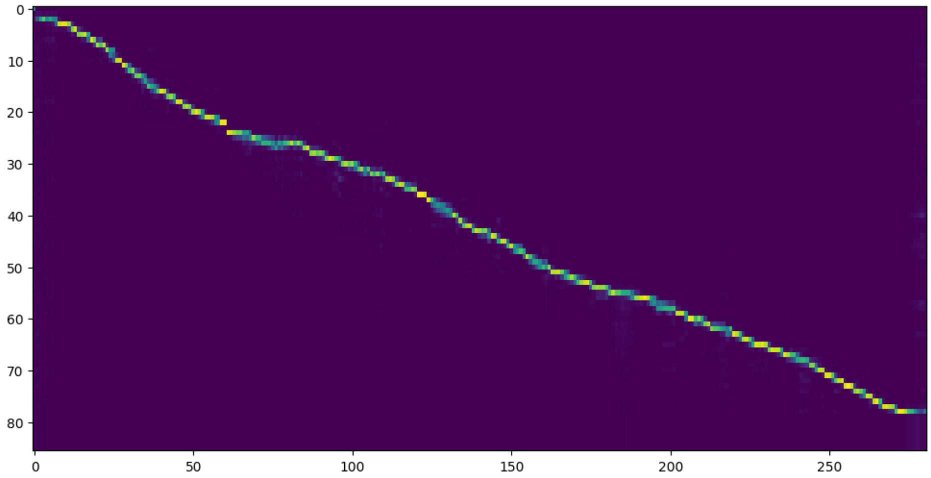


Figure 7: Attention graph for step 210500 of training.

We can see from Figure 7 that the line is clearly visible, so the model did learn attention. From Figure 8 we can see the predicted mel-spectrogram is very close to the target one.

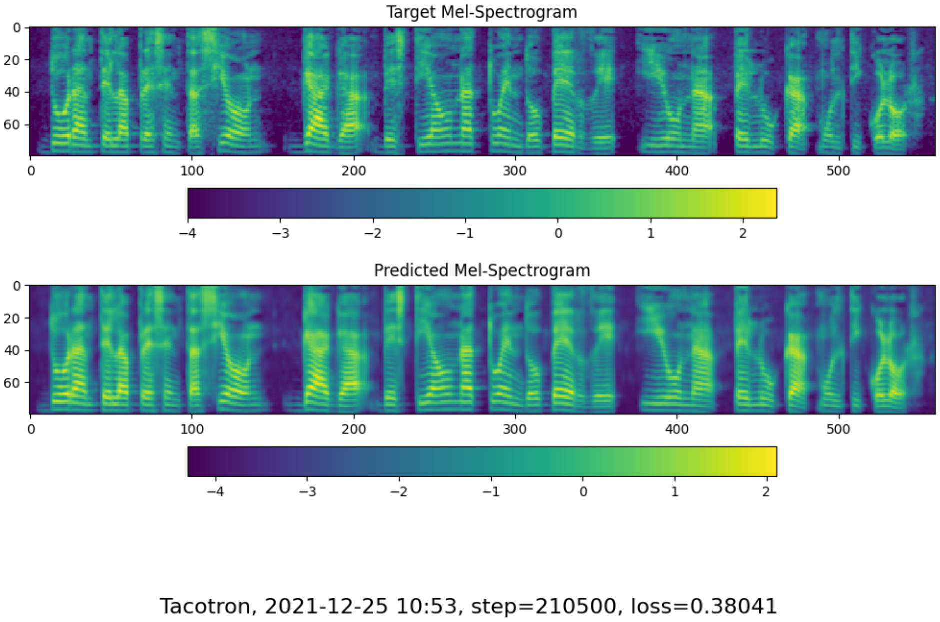
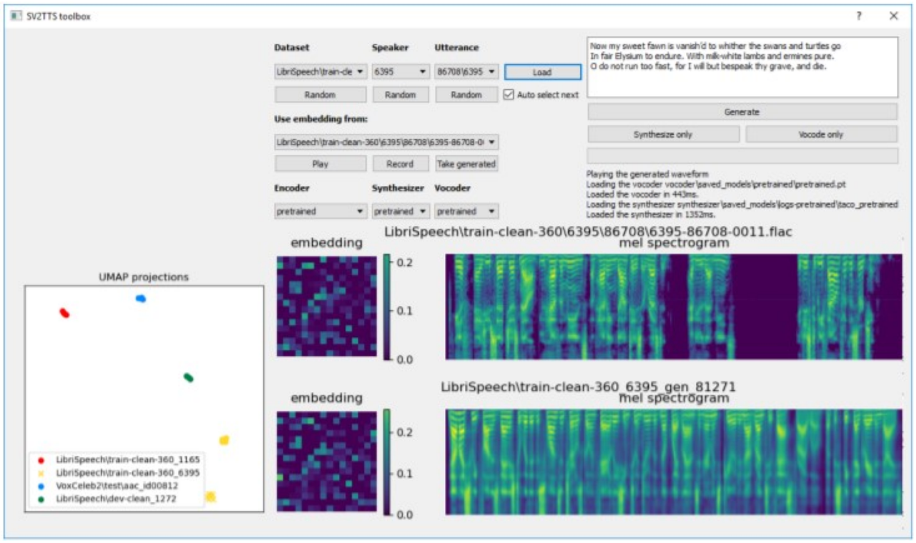


Figure 8: Target and predicted mel-spectrograms for step 210500 of training.

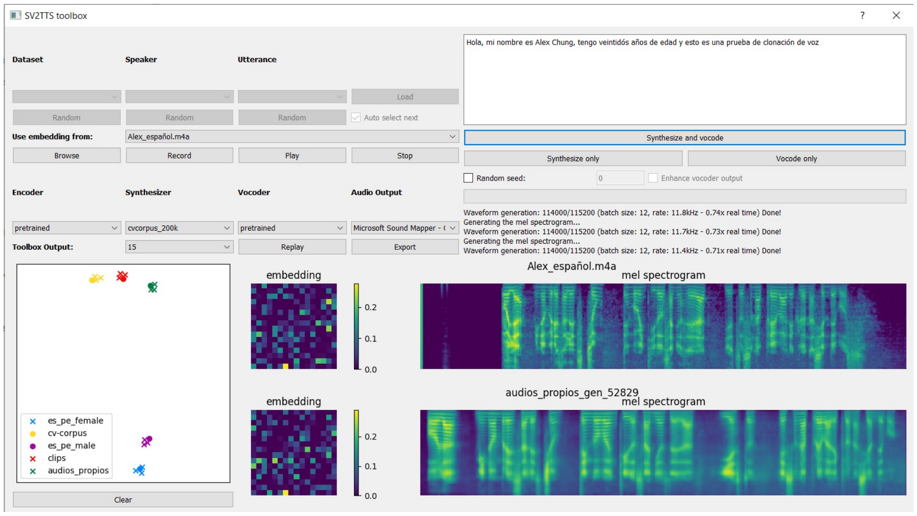
From Figure 7 and Figure 8, we can say that the training succeeded. However, at inference time, there are still some issues: for some utterances, when the produced audio exceeds the length of the original sample, the system fills the excess with noise or silence, and when you put periods as punctuation of the text you want to convert into speech, many of the times they are understood as silences, as if the audio ended in the period. These issues also appear when using racoonML's pretrained model (`https://github.com/raccoonML/Real-Time-Voice-Cloning/releases/tag/Spanish-1`), unfortunately we were run out of time before figuring out a solution for the issues.
Finally, we present the toolbox implemented by Jemine, using spanish speakers utterances to prove the complete framework works well with our synthesizer trained model. We prove this by comparing Figure 9 with Figure 10, where the UMAP projections show us how well the encoder is recognizing the multiple speakers (and their generated audios), while the mel-spectrograms show us the optimal operation of our synthesizer model.
As we can see from our results in Figure 10, the generated mel-spectrogram is smoother than the ground truth, and it also cuts the silences. This a typical behaviour of the model predicting the mean in presence of noise [10]. We can also



Now my sweet fawn is vanish'd to whither the swans and turtles go
In fair Elysium to endure. With milk-white lambs and ermines pure.
O do not run too fast, for I will but bespeak thy grave, and die.

```
Playing the generated waveform
Loading the vocoder vocoder\saved_models\pretrained\pretrained.pt
Loaded the vocoder in 443ms.
Loading the synthesizer synthesizer\saved_models\logs-pretrained\taco_pretrained
Loaded the synthesizer in 1352ms.
```

Figure 9: Jemine's toolbox implementation of the SV2TTS toolbox interface.
Source: Figure 21 [10]



Hola, mi nombre es Alex Chung, tengo veintidós años de edad y esto es una prueba de clonación de voz

```
Waveform generation: 114000/115200 (batch size: 12, rate: 11.8kHz - 0.74x real time) Done!
Generating the mel spectrogram...
Waveform generation: 114000/115200 (batch size: 12, rate: 11.7kHz - 0.73x real time) Done!
Generating the mel spectrogram...
Waveform generation: 114000/115200 (batch size: 12, rate: 11.4kHz - 0.71x real time) Done!
```

Figure 10: Our toolbox results from Jemine's toolbox interface.

notive the embeddings from the ground truth and generated mel-spectrograms are pretty similar, from there we conclude the optimal operation of the model mentioned before, since we used the same text of the target audio to be generated for this example.

5 Conclusion

We achieved good results with the RTVC framework for cloning a spanish speaker's voice unknown during training only by training a model for the synthesizer. Our results are comparable to the english original model. We provide this model to the community in our github repository and also the first github page that compares spanish models' results with the original model. We hope this page can help the community to obtain the MOS for the models beyond the scope of this thesis.

References

[1] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou. Neural voice cloning with a few samples. 02 2018.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

[3] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu. Adaspeech: Adaptive text to speech for custom voice, 03 2021.

[4] C.-M. Chien, J.-H. Lin, C.-y. Huang, P.-c. Hsu, and H.-y. Lee. Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8588–8592, 2021. doi: 10.1109/ICASSP39728.2021.9413880.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[6] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. 2018.

[7] D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32:236 – 243, 05 1984. doi: 10.1109/TASSP.1984.1164317.

[8] A. Guevara-Rukoz, I. Demirsahin, F. He, S.-H. C. Chu, S. Sarin, K. Pipatsrisawat, A. Gutkin, A. Butryna, and O. Kjartansson. Crowdsourcing Latin American Spanish for Low-Resource Text-to-Speech. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, pages 6504–6513, Marseille, France, May 2020. European Language Resources Association (ELRA). ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.801`.

[9] K. Ito and L. Johnson. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[10] C. Jemine. Master thesis : Real-time voice cloning, 2019. URL `https://matheo.uliege.be/handle/2268.2/6801`.

[11] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Lopez-Moreno, and Y. Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *CoRR*, abs/1806.04558, 2018. URL `http://arxiv.org/abs/1806.04558`.

[12] K. Kovitvongsa and P. Lobel. Convenient fish acoustic data collection in the digital age. 12 2021.

[13] S. Liu, D. Su, and D. Yu. Meta-voice: Fast few-shot style transfer for expressive voice cloning using meta learning, 11 2021.

[14] H.-T. Luong. *Deep learning based voice cloning framework for a unified system of text-to-speech and voice conversion.* PhD thesis, 09 2020.

[15] H.-T. Luong and J. Yamagishi. Nautilus: a versatile voice cloning system, 05 2020.

[16] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. 2017.

[17] P. Neekhara, S. Hussain, S. Dubnov, F. Koushanfar, and J. McAuley. Expressive neural voice cloning, 01 2021.

[18] A. 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. 09 2016.

[19] V. Panayotov, G. Chen, and D. Povey. Librispeech: An asr corpus based on public domain audiobooks. 2015.

[20] G. Ruggiero, E. Zovato, L. Di Caro, and V. Pollet. Voice cloning: a multi-speaker text-to-speech synthesis approach based on transfer learning, 02 2021.

[21] J. Shen, R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. Saurous, Y. Agiomyrgiannakis, and Y. Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. 12 2017.

[22] D. Tan, H. Huang, G. Zhang, and T. Lee. Cuhk-ee voice cloning system for icassp 2021 m2voc challenge, 03 2021.

[23] L. Wan, Q. Wang, A. Papir, and I. Moreno. Generalized end-to-end loss for speaker verification, 10 2017.

[24] Y. Wang and R. Skerry-Ryan. Tacotron: Towards end-to-end speech synthesis. pages 4–5, 04 2017.

[25] J. Yamagishi, C. Veaux, and K. MacDonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92), 2019. URL `https://doi.org/10.7488/ds/2645`.

[26] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu. Libritts: A corpus derived from librispeech for text-to-speech, 2019.