Practical Study of Deep Learning Models for Speech Synthesis

Quentin Langlois
quentin.langlois@uclouvain.be
Université Catholique de Louvain
Louvain-la-Neuve, Belgium

Sébastien Jodogne
sebastien.jodogne@uclouvain.be
Université Catholique de Louvain
Louvain-la-Neuve, Belgium

ABSTRACT

Speech synthesis systems, also known as Text-To-Speech (TTS) systems, are increasingly frequent nowadays, with multiple applications such as voice assistants and screen readers for visually impaired or blind people. These applications require strong real-time capabilities to be usable in practice, which can be at the cost of a reduced quality in the synthesized voices. Deep Learning models, which have shown impressive results in the task of audio generation, are hardly ever used for everyday TTS because of their high demand in computational resources. Training such models also requires a large amount of good quality data, which is not available for most languages. This paper explores the benefits of cross-lingual transfer learning, both in terms of training time and amount of data that is needed to obtain good quality models. Our contributions are evaluated with respect to other TTS systems available for the French language. The main observation is that good quality single-speaker models can be trained within half a week on a single GPU, with a limited number of good quality data, by combining transfer learning with few-shot learning.

KEYWORDS

Speech Synthesis, Text-to-Speech, Deep Learning, Transfer Learning, Voice Cloning

ACM Reference Format:
Quentin Langlois and Sébastien Jodogne. 2023. Practical Study of Deep Learning Models for Speech Synthesis. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '23), July 05–07, 2023, Corfu, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3594806.3596536

1 INTRODUCTION

Speech synthesis, or Text-To-Speech (TTS), is a general Natural Language Processing (NLP) task whose goal consists in converting natural text to audio signals. The most common application of such technology is the real-time, automated reading of texts, notably to help visually impaired or blind people to use their electronic devices, for instance to access their e-mails or messages. In this context, the most important criterion is the capability of the audio generation system to run in real-time. The high computational power that is needed to apply Deep Learning (DL) models for TTS is the main reason why such models are not used in practice for everyday audio synthesis. However, this real-time capability often results in poor intonation, fluidity, and *humanity* (i.e., robotic aspect). But such criteria tend to be critical if audio synthesis is used during extended periods of time, such as for document listening. This may notably explain why audio book applications are nowadays only based on human speakers, and never on synthesized voices.

According to the discussion above, this paper first explores multiple aspects of DL-based voice cloning in terms of training time and real-time usability on laptop computers, rather than on powerful dedicated servers. Our work is based on the *Transfer Learning from Speaker Verification to Text-To-Speech* (SV2TTS) architecture [6], because SV2TTS is widely available as free and open-source software with powerful English pre-trained weights. We experimentally show that cross-lingual transfer learning [17] can be used to effectively train good quality models on a new language, in this case French.

Our second contribution is related to few-shot learning. We analyze how transfer learning can be used to quickly train a TTS model for a single speaker with only a limited number of good quality samples, given a multi-speaker TTS model trained with low-quality or noisy data. Many modern zero-shot models, such as VALL-E [20], can clone voices with particularly good similarity and quality, using a single 3-second sample of the voice to copy. However, such models require a large amount of good quality audios, which is not widely available for most languages, including French.

Finally, we introduce Deep Reader, a new audio-book mobile application that uses Deep Learning models to read in real-time any text document that is uploaded to the application. A client-server infrastructure is still needed, as mobile devices may not be powerful enough to locally run the TTS models. Nevertheless, our application requires neither a dedicated, powerful server, nor a cloud infrastructure, as it can be smoothly executed on laptops with a regular Graphics Processing Unit (GPU) with at least 4GB of VRAM. Altogether, this paper highlights promising new practical approaches related to Deep Learning for speech synthesis, notably to the benefit of visually impaired people.

2 RELATED WORK

This section describes related work about the acceleration of training and inference of TTS models. It is important to notice that this paper will only focus on the SV2TTS architecture, not on more recent ones, such as VALL-E [20]. This choice is explained by the fact that pre-trained English models for SV2TTS are available as open data, together with their training time, which eases a comparative evaluation. Nevertheless, the methodology adopted by our work could be similarly investigated on other TTS architectures.

PETRA '23, July 05–07, 2023, Corfu, Greece

ACM ISBN 979-8-4007-0069-9/23/07…$15.00
https://doi.org/10.1145/3594806.3596536

2.1 Transfer Learning

Transfer learning [22] is a general method that is used in multiple domains, and that consists in using the weights of a model trained on a task A, as the initialization to train a new model on a new task B (this procedure is commonly referred to as *fine-tuning*). Transfer learning has been widely applied to Computer Vision (CV) and Natural Language Understanding (NLU) tasks, but less to audio-based models. Indeed, the benchmarking of TTS models is mostly done in English, which reduces the scientific interest in the development of cross-lingual transfer learning. However, the latter approach may have a huge interest to accelerate the training of good TTS models in other languages, which is especially important for languages with lower availability of data [17]. Our work accordingly explores the training of a multi-speaker French TTS architecture through cross-lingual transfer learning based on an English pre-trained model. Then, the French model is fine-tuned on few good-quality data from a single speaker. This technique is known as *few-shot learning*.

2.2 SV2TTS for Speech Synthesis

Transfer Learning from Speaker Verification to Multi-speaker Text-To-Speech Synthesis (SV2TTS) [6] is a Deep Learning architecture that is based on three separate modules: (1) a *Speaker Encoder* that maps one audio sample onto a d-dimensional vector (referred to as the *audio embedding*), (2) a *Synthesizer* that combines the text with the audio embedding, to generate the corresponding *mel-spectrogram* (the mel-spectrogram is a non-linear, non-revertible transformation of the regular frequency spectrogram), and (3) a *Vocoder* that transforms the mel-spectrograms into an audio sample.

Figure 1 depicts the SV2TTS pipeline[1]. An important aspect is that each of the three modules of the SV2TTS pipeline corresponds to one DL model, and that these three models are trained separately from each other. The three modules will now be successively reviewed in more detail.

2.2.1 Speaker Encoder. The architecture of the Speaker Encoder (SE) module used in the SV2TTS paper corresponds to a Long Short-Term Memory (LSTM) network [3] with 3 hidden layers that is trained to discriminate the audio samples from different speakers [19]. The general goal of the Speaker Encoder is to generate embeddings from given audio samples, so that all the embeddings of one speaker are close to each other in this vector space, while the embeddings from different speakers are distant from each other.

More precisely, the Speaker Encoder network is trained to minimize the Generalized-End-to-End (GE2E) loss function [19]. The GE2E loss function was initially designed to speed up and improve the performance metrics when training Deep Learning models for the task of Speaker Verification [2]. The general principle behind GE2E is to encode N audios from M different speakers at once (i.e., the training batch size is $M \times N$), then to compute the average embedding for each speaker (named the *speaker centroid*), and finally to reduce the distance between each audio embedding and its corresponding speaker centroid. This procedure is illustrated in Figure 2.

2.2.2 Synthesizer. The Synthesizer used in the SV2TTS architecture is based on the general Tacotron-2 architecture [14]. The Tacotron-2 model is made of two components: (1) a text encoder, that encodes the input embedded text using a 1-dimensional Convolutional Neural Network (1D-CNN) architecture coupled with a Bidirectional LSTM, and (2) an auto-regressive decoder with an attention mechanism, which produces the mel-spectrogram for the input text.

The multi-speaker extension of Tacotron-2 concatenates the output vectors of the text encoder (i.e., the output of the Bidirectional LSTM layer) with the d-embedding vector of any audio sample of the voice to be cloned, as encoded by the Speaker Encoder. Thanks to this adaptation, the auto-regressive decoder has access to both the text to read, and to the way the text should be read.

2.2.3 Vocorder. The SV2TTS Vocoder is Waveglow [12], a non-auto-regressive architecture inspired by Glow [7] and WaveNet [18], which is mainly composed of invertible 1-dimensional convolutional layers. Waveglow has been trained to be a *general Vocoder*, meaning that its objective task is to maximize the probability of the waveform, while not being specific to a single speaker. This model can therefore be applied to convert any mel-spectrogram to its corresponding audio sample, no matter the speaker or the language.

2.3 Siamese Networks

Besides transfer learning and SV2TTS, our contributions also resort to the concept of Siamese Networks. Indeed, the main goal of the Speaker Encoder part is to create an embedding vector that hopefully characterizes the speaker's voice. To this end, SV2TTS uses the GE2E-Loss function, which necessitates to have a batch size that is as large as possible, because the loss is computed on the similarity between all the inputs within each batch. However, other methods do exist to compute such embeddings using less computational power and less memory. In particular, the so-called *Siamese Network* (or *Twin Network*) architecture [8] is made of a single model that encodes two inputs independently, which produces two d-dimensional embedding vectors, and then computes a distance metric between the produced embeddings. If the pair comes from the same label (in the case of the Speaker Verification task, from the same speaker), the distance is minimized, and maximized otherwise.

3 METHODS

This section introduces the multiple refinements that have been performed on the aforementioned Deep Learning architectures, to speed up their training time and their associated performance. It is important to notice that the Vocoder module of the architecture has not been changed. Indeed, since the open, pre-trained implementation of Waveglow is a general Vocorder, it can be applied to any mel-spectrogram without fine-tuning. Furthermore, general Vocorder models tend to be very large, making them challenging to train on regular computers.

[1] Note that the standard SV2TTS architecture does not imply any fine-tuning from a pre-trained model: The "transfer learning" term that is part of the name of SV2TTS instead refers to a transfer from the Speaker Encoder to the Synthesizer, which enables the multi-speaker feature.

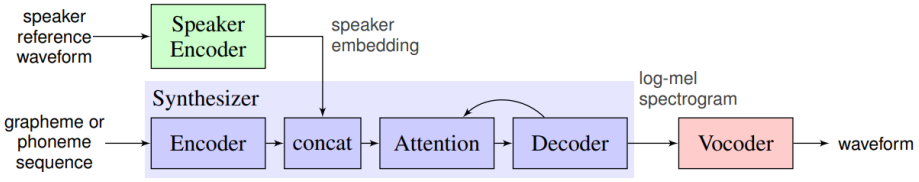

Figure 1: Overview of the SV2TTS architecture (reproduced from [6]).

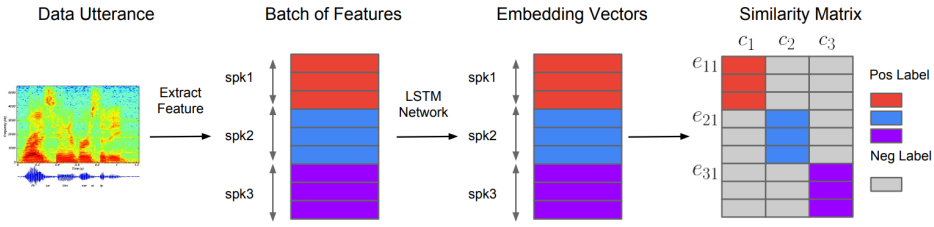

Figure 2: Computation of the similarity between samples and its corresponding speaker centroid (reproduced from [19]).

3.1 Refining the Speaker Encoder

This section describes the adaptations that were made to the Speaker Encoder (SE) module from the original SV2TTS paper. The SE converts the input audio into a hopefully meaningful representation (the embedding), that will be subsequently used as the input for the Synthesizer module. As explained in Section 2.2.1, the original SE corresponds to the LSTM model with 3 hidden layers that was originally introduced by Wan et al. in their paper about the GE2E loss function [19].

3.1.1 Underlying Deep Learning Model. We propose to replace the 3-layers LSTM model with a simpler architecture that is based on a 1-dimensional Convolutional Neural Network (1D-CNN). The 1D-CNN is used to downsample the input mel-spectrogram (of variable length). The 1D-CNN is followed by a LSTM, then by a L2-normalization (as illustrated in Figure 3). This downsampling by the 1D-CNN has been introduced to limit the recursion steps that were required by the LSTM layers of the original SV2TTS as much as possible. Furthermore, thanks to the parallelization capabilities of CNN networks, the training time is vastly improved compared to the original 3-layers LSTM model.

3.1.2 Siamese Networks. The main difficulty in training Siamese Networks is to correctly define the distance metric that is used to compare the produced embeddings, and to choose the correct loss function for distance minimization/maximization. To deal with this challenge, we propose to add a 1-neuron Fully Connected layer with a sigmoid activation function at the end of the network, that takes as input the distance between the pair of embeddings, which is illustrated in Figure 4. Classically, the distance metric would either be the dot product or the cosine similarity, but our experiments indicate that the Euclidian distance gives faster convergence while preserving reliable results. This allows to rephrase the attraction/repulsion principle underpinning Siamese Networks as a straightforward binary classification task with two possible outputs (*same* or *different*). This refinement allows to train the model with the regular negative log-likelihood loss, and to directly have a probability score between 0 and 1 for Speaker Verification, in the place of a non-normalized distance or similarity score.

As this model compares pairs of inputs, it is common to have vastly unbalanced data, because pairs of audio samples from different speakers are much more present than pairs from the same speaker. In order to have a balanced training procedure, we propose to randomly sample, at each step, N audios from a given speaker, and pair them with $N/2$ audios from another speaker and with $N/2$ audios from the same speaker, while preventing such pairs to hold the same audio sample twice. This sampling procedure enforces a balanced distribution of the two labels. Pre-loading some audios allows a huge speed up in the training procedure by avoiding redundant access to the filesystem, as multiple audios may be used in multiple pairs.

3.1.3 GE2E Loss Data Generator. The original GE2E loss function has not been changed in this work. Nevertheless, this loss function can be used in multiple ways in practice. We propose to introduce *data augmentation* in the generation pipeline, as a way to speed up the training and to increase the robustness of the resulting model.

More precisely, the principle of GE2E is to have N audios from M speakers at each batch, and to compute the similarity matrix using the speakers' centroids. The proposed strategy is to pre-sample, for each speaker, N audio files, which allows to have all the samples for a given loop over the dataset. This simple preprocessing loop can be repeated multiple times, allowing to have lots of pairs, and a representative sampling for each speaker. An interesting consequence is that this process creates data augmentation by design. Indeed, the same audios are sampled multiple times across the sampling loop (especially for speakers with fewer data), but appear in multiple contexts (i.e., paired with many different samples or speakers), which prevents overfitting and increases robustness. To

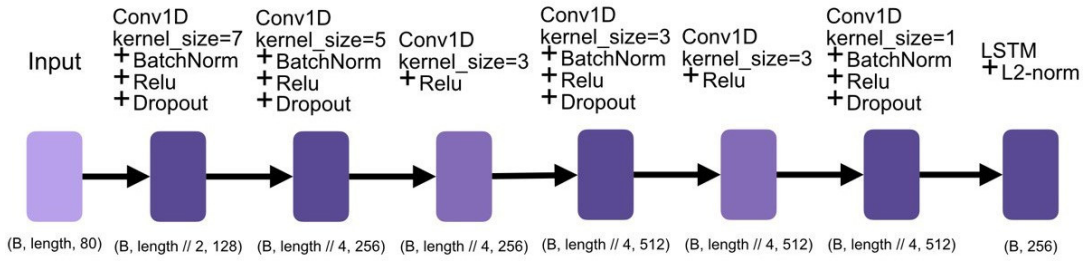

Figure 3: 1D-CNN Encoder architecture used in both Siamese Network model and GE2E-based encoder.

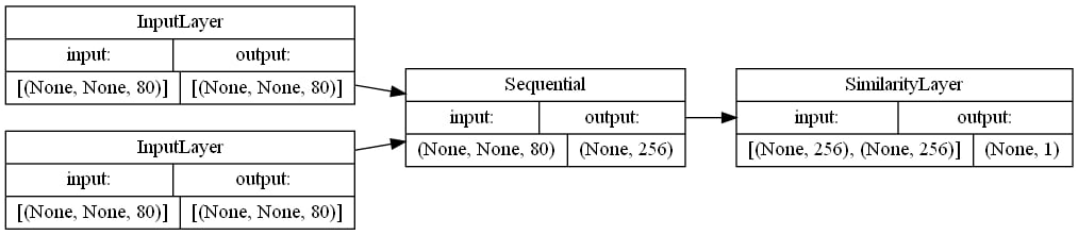

Figure 4: Variant of the Siamese Network: The `SimilarityLayer` computes the distance metric (e.g. Euclidian), and is followed by the 1-unit Fully Connected layer with sigmoid activation.

further exploit these benefits, data can be shuffled between each epoch, so that the speakers' samples do not change, but are paired with different speakers at each epoch.

3.2 Refining the Synthesizer

This section discusses the adaptations that were brought to the Tacotron-2 architecture corresponding to the Synthesizer module. The model that is provided by NVIDIA on LJSpeech [4] served as a pre-trained Tacotron-2 model. The same model architecture and processing pipeline have been used to fully perform transfer learning. This architecture is however inherently focused on a single speaker. This raises the question of the effective training of a multi-speaker model for the Synthesizer.

As explained in Section 2.2.2, the multi-speaker extension of Tacotron-2 involves a concatenation between the output vectors of the text encoder, with the voice embedding. Because of this concatenation, the shape of the inputs of the auto-regressive decoder of Tacotron-2 module has changed. Consequently, the shape of the weight matrices has also changed compared to the single-speaker architecture, which prevents a direct transfer. To solve this issue, we propose to introduce the new concept of *partial transfer learning*. This procedure consists in transferring only the common parts of the matrices of weights. Our experiments have shown that transferring the common part, and initializing the remaining weights to zero, gives a much faster training with respect to simply skipping the layers whose matrices of weights have a different shape.

4 RESULTS

The multiple refinements that were introduced in Section 3 are now evaluated. In our experiments, both the Speaker Encoder and the multi-speaker Synthesizer are trained on the Common Voice French dataset [1], the VoxForge database [9], and the SIWIS French Database [21]. The reference training time is 4 days on 4 GPU for the Speaker Encoder model, and 150k steps with a batch size of 144 across 4 GPU for the Synthesizer model [5]. The original pre-trained Tacotron-2 English model comes from the NVIDIA public repository [10]. The evaluation scripts, the audios and the evaluation results have been published on GitHub[2].

4.1 Training

Table 1 lists information about the training of the various modules that are used in the SV2TTS architecture. The *pre-training* column shows the pre-trained model used as initialization, meaning that the two models with the *English* model are pure English-to-French cross-lingual models, while the others are fine-tuned versions of these cross-lingual models for French.

The two first models in Table 1 refer to the audio encoders used in the SV2TTS models. Their embedding dimension is 256. The *Multi* refers to a multi-speaker model using either the *Siamese* or the *GE2E-based* encoders, *Female* refers to a model fine-tuned on the SIWIS database, and *Male* refers to a model trained on the Kaggle Single Speaker database [11]. The number after *Male* or *Female* corresponds to the number of data used for training, which is used to distinguish between the models in the following sections.

For the multi-speaker model using the GE2E-based SE (*Multi GE2E (1)*), a first training round has been performed with only 37k audios (around 800 speakers) for 75 epochs (37k steps lasting 28 hours). This first round aims to make the model learn the French language. The second round of 15 epochs uses more than 260k audios from 2000 speakers with at most 2500 samples per speaker. The aim of the second round is to improve the cloning capabilities

[2]Link to the GitHub repository: https://github.com/Ananas120/nlp4disability-tts

Models	Pre-training	Samples	Epochs	Batch size	Training time	GPU
GE2E encoder	/	435712	25	256	4h 46min	3090 Ti
Siamese	/	107520	50	32^3	8h 51min	3080 Max-Q
Multi Siamese	English	74592	25	48	78h 52min	3090 Ti
Multi GE2E (1)	English	36912	75	48	27h 56min	3090 Ti
Multi GE2E (2)	Multi GE2E (1)	266544	15	48	59h 8min	3090 Ti
Male-100	Multi GE2E (2)	104	125	8	1h 28min	3090 Ti
Male-1000	Multi GE2E (2)	1008	50	16	2h 55min	3090 Ti
Female-24	Multi GE2E (2)	24	150	8	45min	3090 Ti
Female-250	Multi GE2E (2)	256	150	16	2h 8min	3090 Ti
Female-3000	Multi Siamese	2848	9	32	2h 21min	3080 Max-Q

Table 1: Information about the training of the French models.

of the model by increasing the number of speakers and accents, and to avoid overfitting on over-represented speakers.

The two multi-speaker models have not been evaluated without fine-tuning because of their high instability in the generation of the mel-spectrograms. This is due to the poor audio quality of the used datasets that feature background noise, bad recording conditions, and long silences. Pre-processing using the `noisereduce` library [13] and silence trimming techniques (by thresholding the audio signal) was implemented to reduce the noise and the moments of silence as much as possible. However, this pre-processing did not solve all the audio quality issues.

The exhaustive list of training hyperparameters is available on GitHub. The number of epochs has been tuned to stop after convergence for the encoders, to stop after a given amount of time for the multi-speaker models, and experimentally tuned to give the best quality results for the single-speaker models. The batch size has been set to fully use the available GPU memory size, except for the single-speaker fine-tuning, where experiments have shown that reducing its size is needed for convergence when less data is available.

4.2 (Comparative) Mean Opinion Score

The Mean Opinion Score (MOS) and Comparative Mean Opinion Score (CMOS) are the classical metrics to evaluate the quality of synthesized voices. They are both performed by human raters. The MOS metric ranges from 0 to 5 with a step of 0.5. MOS evaluates the overall quality of a single read sentence. On the other hand, the CMOS metric ranges from -3 to 3 with a step of 0.5. CMOS compares the relative quality of the read utterance to a reference version (with the same sentence and speaker). A score of -3 means that the reference is much better than the utterance to evaluate, and a score of 3 means the opposite.

Table 2 lists the values of the MOS and CMOS metrics for the various models, after an evaluation by 7 French-speaking volunteers. In this table, the reference is the Ground Truth (GT) mel-spectrogram inverted by the Vocoder model (i.e., through the chain "human audio → mel-spectrogram → Vocoder → audio"). The reason is that the Vocoder has not been fine-tuned, which implies that if it fails to correctly invert the mel-spectrogram of the original sample, the synthesized mel-spectrogram will never reach a better audio quality.

Model	MOS	CMOS
Male	3.96 (+/- 0.46)	2.13 (+/- 0.46)
Male (GT)	2.62 (+/- 0.49)	0
Male-100	1.83 (+/- 0.64)	-1.03 (+/- 0.61)
Male-1000	2.04 (+/- 0.76)	-0.65 (+/- 0.82)
Female	3.84 (+/- 0.31)	0.47 (+/- 0.44)
Female (GT)	3.59 (+/- 0.26)	0
Female-3000	3.07 (+/- 0.48)	-0.64 (+/- 0.56)
Female-24	2.97 (+/- 0.53)	-0.67 (+/- 0.61)

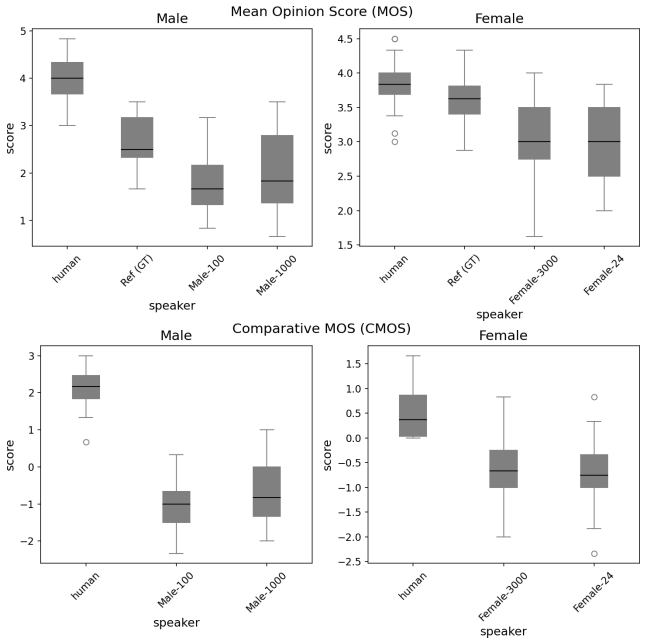

Table 2: Evaluation results for the MOS and CMOS metrics, both in tabular and box-plot forms.

According to this table, the Vocoder is not able to correctly invert the male voice (cf. the significant difference between the human and GT scores), probably because of a deeper voice. Indeed, inverted male voices tend to have a background noise, unlike female voices. A

hypothesis to explain this observation is that the pre-trained version of WaveGlow is not able to correctly invert the lower frequencies that are more present in male voices. Nevertheless, the trained synthesizers tend to have 1 point lower than the GT for both the male and the female speakers. According to these metrics, the male voice is around 3 points higher than the synthesized voices, and 1 point higher for the female voice. However, the MOS and CMOS metrics do not tell whether a synthesized voice is pleasant or not, nor whether it is usable in a real-world application or not.

4.3 Long-term Mean Opinion Score

The main drawback of the MOS/CMOS metrics is that voices are evaluated on single sentences, not on longer texts or on the long-term *pleasantness* to listen to the voice. This is however an important criterion for document-reading usage of such technologies.

In this context, we propose to introduce the Long-term Mean Opinion Score (LMOS), with the goal to evaluate and compare voices on a single long text with a duration between 1 and 2 minutes of audio. The goal is to evaluate neither the overall quality of the voice, nor its similarity with the cloned voice (which is evaluated by MOS/CMOS), but to assess whether a voice is pleasant or not, and whether a user may listen to an entire document (like a book) with this voice. We accordingly define LMOS as a qualitative metric whose score ranges from 0 to 5 with a step of 0.5, and whose values are defined in Table 3. The main difference with the MOS metric is that a human voice may not have a good LMOS score if its rhythm or prosody is unpleasant, while in MOS it is nearly impossible that a model does better than its corresponding human voice based on a single sentence. The main advantage of the LMOS metric is that the evaluation criterion is well defined and can be used to assess the usability of a voice in speech synthesis applications. Furthermore, the long-term aspect allows one to ignore some sparse errors in pronunciation or rhythm, which are much more damaging to metrics defined on isolated sentences.

An excerpt from the French novel "*20 000 lieues sous les mers*" by Jules Verne was used to make a LMOS-based comparison between the models trained in this paper, a human reader, and two freely available online speech synthesis systems for French, namely TTSFree [15] and TTSMP3 [16][4]. 9 French-speaking volunteers took part in the LMOS ranking[5]. The results of this evaluation are reported in Table 4. These results indicate that, for long-term usage, models can be as pleasant to listen to than the human speaker. However, this remains a subjective evaluation based on personal appreciations. This subjective criterion is nevertheless the most relevant, as far as document reading is concerned.

4.4 The DeepReader Real-World Application

This section aims to discuss some potential usages of TTS systems based on Deep Learning, despite their relative slowness and higher power requirements compared to regular speech synthesis systems. To this end, a pilot application named DeepReader was developed.

[4]Both TTSFree and TTSMP3 are proprietary, closed-source systems. It is thus impossible to know what are the exact algorithms these two systems leverage. However, by observing the generation time and the quality of generated audio, TTSFree is most probably based on DL technologies, whereas TTSMP3 most probably uses regular speech synthesis algorithms.

[5]Out of these 9 volunteers, 7 also evaluated MOS and CMOS.

Value	Description
0	Badly read and not pleasant to listen to.
3	Well read but not pleasant to listen to.
4+	Not perfectly read but still pleasant to listen to. I could listen to the whole document with this voice.
5	Perfectly read and very pleasant.

Table 3: Definition of the LMOS scale.

	LMOS
Female-250	4.14 (+/- 0.69)
Human	4.06 (+/- 0.60)
TTSFree	3.71 (+/- 0.75)
Female-3000	3.56 (+/- 0.98)
TTSMP3	3.36 (+/- 1.09)
Female-24	3.28 (+/- 0.58)
Male-1000	2.56 (+/- 0.76)
Male-100	2.33 (+/- 1.00)

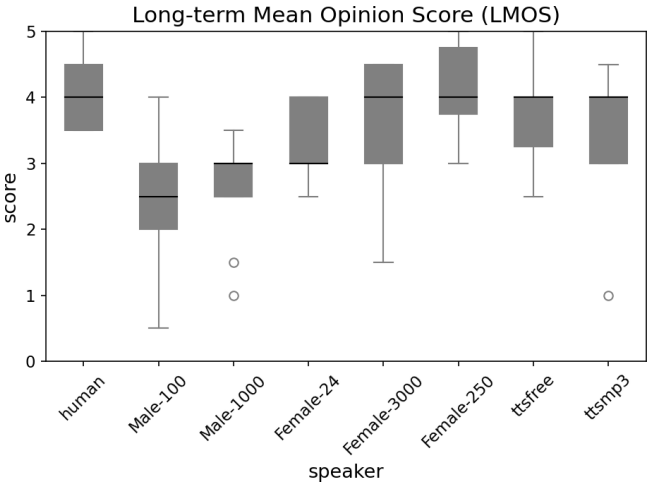

Table 4: Evaluation results for the LMOS metric, both in tabular and box-plot forms.

This application is designed to read documents in real-time using the TTS models detailed in the previous sections.

The TTS pipeline that uses the SV2TTS and WaveGlow models are not usable for real-time speech synthesis on low-end devices, notably on mobile devices. This is the reason why DeepReader adopts a client-server software architecture. Nevertheless, the complete inference pipeline can generate 3 seconds of audio in 1 second on an RTX3090 Ti GPU. This aspect allows one to use SV2TTS models in contexts where the real-time ability is not a strict requirement, which is typically the case of audio books, for they are typically pre-recorded. In the DeepReader system, audios are generated while the user is listening to the earlier paragraph, which allows to listen without any interruption.

DeepReader is therefore similar to regular audio book applications, except that it can read any kind of document (it only requires

a proper processing function to extract the text), and does not require any human intervention, which allows to directly listen to the document posted on the server. This may allow new potential usages of Deep Learning models for Text-to-Speech, for anyone who prefers to listen to a document instead of reading it. Some examples include students with an auditory memory who prefer to listen to their notes or course syllabi, illiterate people, as well as people who want to learn a new language by listening to the pronunciation of a native speaker.

5 DISCUSSION AND CONCLUSION

This paper shows that cross-lingual transfer learning allows a large speed up in the training time for the SV2TTS architecture. Thanks to the few-shot learning procedure, such models can produce near-human quality audio within a week of training on single GPU, with only a limited amount of good quality audio from a single speaker. The important aspect is that this technique can be generalized to any other language for which a large amount of poor quality data and a limited amount of good quality data are available.

The Long-term Mean Opinion Score (LMOS) introduced in this paper seems to be highly correlated with the MOS and CMOS metrics, while evaluating the different speech synthesis systems by taking the global quality of the generated audio over time into consideration. Furthermore, this metric is faster to evaluate, as it only requires listening to a single audio of 2 minutes. This contrasts with MOS and CMOS that typically require listening to multiple audios of 5 to 10 minutes, which is much more time consuming and displeasing for the evaluators. The aim of LMOS is not to replace MOS and CMOs, as the criteria differ, but to introduce new practical, complementary evaluation criterion for speech synthesis systems.

In this work, we have only re-trained the Speaker Encoder and the Synthesizer modules, because Vocoders tend nowadays to become increasingly general (i.e., usable on any voice). However, as shown with the male voice, it may still fail and introduce background noise, which degrades the overall quality and pleasantness of the synthesized audios. Therefore, further work could tackle the fine-tuning of such models, at the expense of longer training time and increased computational power. Finally, it is worth noticing that Deep Learning voice cloning technologies can also be helpful to a general audience, beyond the specific case of visually impaired people, for instance for the automated generation of podcasts or audio books, for people with an auditory memory, for illiterate people, or for people learning foreign languages.

ACKNOWLEDGMENTS

The authors would like to thank Celia Hassaine, Felix Gaudin, Julia Franken, Amaury Fierens, François de Keersmaeker, Gorby Kabasele Ndonda, and the others, for their help in evaluating the different models. The DeepReader application was co-developed by Mohammad Hassan Zareie, who also evaluated the models.

REFERENCES

[1] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2019. Common Voice: A Massively-Multilingual Speech Corpus. *CoRR* abs/1912.06670 (2019), 5 pages. http://arxiv.org/abs/1912.06670

[2] Ujwala Baruah, Rabul Hussain Laskar, and Biswajit Purkayashtha. 2020. Speaker Verification Systems: A Comprehensive Review. In *Smart Computing Paradigms: New Progresses and Challenges*, Atilla Elçi, Pankaj Kumar Sa, Chirag N. Modi, Gustavo Olague, Manmath N. Sahoo, and Sambit Bakshi (Eds.). Springer Singapore, Singapore, 195–207.

[3] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9 (1997), 1735–1780.

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

[5] Corentin Jemine. 2019. *Real-Time Voice Cloning*. Master's thesis. ULiège. https://matheo.uliege.be/handle/2268.2/6801

[6] Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. 2018. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. *CoRR* abs/1806.04558 (2018), 15 pages. arXiv:1806.04558 http://arxiv.org/abs/1806.04558

[7] Diederik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. *ArXiv* abs/1807.03039 (2018), 15 pages. https://arxiv.org/abs/1807.03039

[8] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese Neural Networks for One-Shot Image Recognition. In *ICML deep learning workshop*, Vol. 2. W&CP, Lille, 8 pages.

[9] Ken MacLean. 2018. Voxforge. http://www.voxforge.org/home

[10] NVIDIA. 2020. Tacotron 2 (without wavenet). https://github.com/NVIDIA/tacotron2

[11] Kyubyong Park and Tommy Mulc. 2018. French Single Speaker Speech Dataset. https://www.kaggle.com/datasets/bryanpark/french-single-speaker-speech-dataset

[12] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2018. WaveGlow: A Flow-based Generative Network for Speech Synthesis. *CoRR* abs/1811.00002 (2018), 5 pages. http://arxiv.org/abs/1811.00002

[13] Tim Sainburg. 2019. `timsainb/noisereduce` v1.0. https://doi.org/10.5281/zenodo.3243139

[14] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2017. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. *CoRR* abs/1712.05884 (2017), 5 pages. http://arxiv.org/abs/1712.05884

[15] `TTSFree.com`. 2023. TTSFree – Free Text-To-Speech and Text-to-MP3 for French. https://ttsfree.com/text-to-speech/french/

[16] `TTSFree.com`. 2023. TTSMP3 – Free Text-To-Speech and Text-to-MP3 for French. https://ttsmp3.com/text-to-speech/French/

[17] Tao Tu, Yuan-Jui Chen, Cheng-chieh Yeh, and Hung-yi Lee. 2019. End-to-end Text-to-speech for Low-resource Languages by Cross-Lingual Transfer Learning. *CoRR* abs/1904.06508 (2019), 5 pages. http://arxiv.org/abs/1904.06508

[18] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. *CoRR* abs/1609.03499 (2016), 15 pages. http://arxiv.org/abs/1609.03499

[19] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. 2017. Generalized End-to-End Loss for Speaker Verification. https://doi.org/10.48550/ARXIV.1710.10467

[20] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. https://doi.org/10.48550/ARXIV.2301.02111

[21] Junichi Yamagishi, Pierre-Edouard Honnet, Philip Garner, and Alexandros Lazaridis. 2017. The SIWIS French Speech Synthesis Database. https://doi.org/10.7488/DS/1705

[22] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2019. A Comprehensive Survey on Transfer Learning. *CoRR* abs/1911.02685 (2019), 31 pages. http://arxiv.org/abs/1911.02685