Université de Liège

Faculté des Sciences Appliquées

Automatic Multispeaker Voice Cloning Across Languages

Author:
Corentin Jemine

Supervisor:
Prof. Gilles Louppe

Academic year 2018 - 2019

Graduation studies conducted for obtaining the Master's degree in Data Science by Corentin Jemine

Possibly, start with a TTS lexicon with some definitions. Meanwhile, I'll make a list of TTS-specific words that may be worth explaining:
coarticulation
linguistic context
spectral envelope
fundamental frequency
contour (a way in which something varies especially the pitch of music or the pattern of tones in an utterance.)
supra-segmental
grapheme

1 Abstract

To do when I'll have a good overview of the project. Try to answer:

- What is the goal of the application? What are its requirements, what is the setting, what kind of data are we going to use it on?
- What is zero-shot voice cloning? How does it fit in here (difference between an online and offline approach)?
- What are the particularities of our implementation (both model and datasets), what are its upsides and downsides (for example: requires huge datasets but fast inference)?
- What did we ultimately achieve? How good are our results?

2 Introduction

2.1 Problem definition

Update this section to reflect what was actually achieved and is presented in this document

We aim to develop a framework that, given an audio segment of speech in a source language from different speakers, a transcript of this segment and a translation of this transcript in a target language, is able to regenerate the audio segment as if each speaker was talking in the target language. This task is to be performed online by the framework but is expected to output an audio of better quality given a longer delay from the source, such that the best performance is obtained when running offline. The quality of an audio sample transferred in a different language relies on the naturalness of the output voice and a meaningful transfer of the speaker features (e.g. someone speaking slowly in French is expected to also speak slowly in English).

2.2 Statistical parametric speech synthesis

Statistical parametric speech synthesis (SPSS) refers to a group of data-driven TTS synthesis methods that emerged in the late 1990s. In SPSS, the relation between features computed on the input text and output acoustic features is modeled by a statistical generative model (called the acoustic model). A complete SPSS framework thus also includes a pipeline to extract features from the text to synthesize as well as a system able to reconstruct an audio waveform from the acoustic features produced by the acoustic model (such a system is called a vocoder). Unlike the acoustic model, these two parts of the framework may be entirely engineered and make use of no statistical methods. If it is possible to condition parts of the framework in such a way that the characteristics of the generated voice are modified, then the framework is a multispeaker TTS synthesis system.

The processing of text into features can be nearly inexistent as it can be very extensive. Speech is an intricate process that depends on a wide range of linguistic contexts. Providing these contexts greatly reduces the extent of the task to be learned by the acoustic model, but may require complex natural language processing (NLP) techniques or accuracy trade-offs, especially for rare or unknown words. Linguistic contexts are retrieved on different levels: utterance, phoneme, syllable, word and phrase. For each of those elements, their neighbouring elements of the same level are usually considered, as well as the elements lower in the hierarchy it comprises. For example, a given frame will contain a word, the two previous words, the two following words and the syllables contained in all those words. The position of each element with regard to its parent element can be included (e.g. fifth word in a sentence), as well as grammatical information such as part of speech. For syllables, the lexical stress and accent can be predicted by a statistical model such as a decision tree. For prosody, ToBI (Beckman and Elam, 1997) is often used. Linguistic contexts are often represented and concatenated into a single vector in order to be exploitable by statistical models. Categorical features are encoded using one-hot representations. It is common for the resulting vector to contain hundreds of values.

Talk about evaluation metrics (mainly MOS)? 50 features? Arpabet? Remove last segment on linguistic contexts?

2.3 State of the art in multispeaker TTS

Previous state of the art in SPSS includes hidden Markov models (HMM) based speech synthesis (Tokuda, 2013). The speech generation pipeline is laid out in figure 1. In this framework, the acoustic model is a set of HMMs. The input features are rich linguistic contexts. Ideally, one would train an HMM for each possible context; but as the number of contexts increases exponentially with the number of factors considered, it is not practical to do so. Indeed, not every context will be found in a typical dataset and the training set would then have to be partitioned over the different contexts, which is data inefficient. Instead, contexts are clustered using decision trees and an HMM is learned for each cluster (Yoshimura et al., 1999). Note that this does not solve entirely the training set fragmentation problem. The HMMs are trained to produce a distribution over mel-frequency cepstral coefficients (MFCC) with energy (called static features), their delta and delta-delta coefficients (called dynamic features) as well as a binary flag that indicates which parts of the audio should contain voice. This is shown in figure 2. A new sequence of static features is retrieved from these static and dynamic features using the maximum likelihood parameter generation (MLPG) algorithm (Tokuda et al., 2000). These static features are then fed through the MLSA vocoder(Imai, 1983). It is possible to modify the voice generated by conditioning on a speaker or tuning the generated speech parameters with adaptation or interpolation techniques (Yoshimura et al., 1997) elaborate a bit on these techniques?, making HMM-based speech synthesis a multispeaker TTS system. Compare with concatenative see (Zen et al., 2013) and ieeexplore.ieee.org/document/541110.

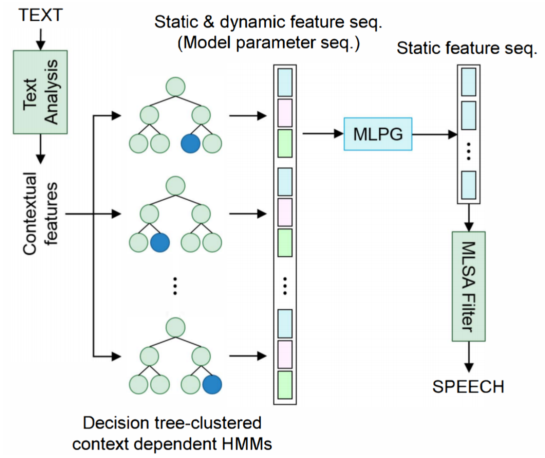


Figure 1: The general HMM-based TTS synthesis approach.

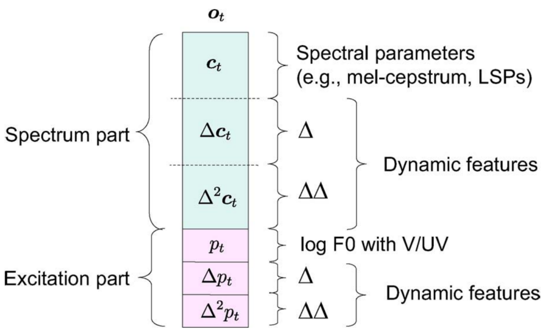


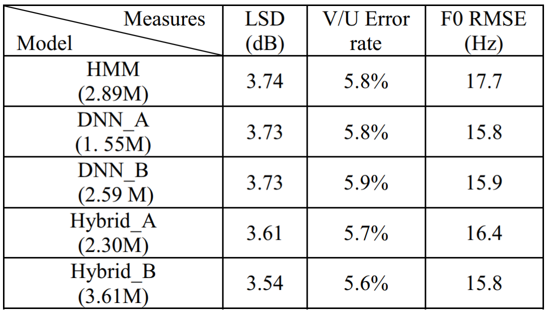
Figure 2: Dynamic and static features produced by the HMMs. F0 is the fundamental frequency and V/UV is the voicing flag.

Improvements to this framework were later brought by feed-forward deep neural networks (DNN), as a result of progress in both hardware and software. (Zen et al., 2013) proposes to replace entirely the decision tree-clustered HMMs in favor of a DNN. They argue for better data efficiency as the training set is no longer fragmented in different clusters of contexts, and for a more powerful model?. They demonstrate improvements over the speech quality with a number of parameters similar to that of the HMM-based approach. Their best model is a DNN with 4 layers of 256 units using a sigmoid activation function. Subjects assessing the quality of the generated audio samples report that the DNN-based models produces speech that sounds less muffled than that of the HMM-based models. Later researches corroborate these findings (Qian et al., 2014). (Hashimoto et al., 2015) additionally studies the effect of replacing MLPG with another DNN. The combinations of HMM/DNN and MLPG/DNN give rise to four possible frameworks, the novel ones being HMM+DNN and DNN+DNN[1], while HMM+MLPG and DNN+MLPG are the frameworks described respectively in (Tokuda, 2013) and (Zen et al., 2013). Each DNN they use is 3 layers deep with 1024 units using a sigmoid activation function. MOS results confirm that DNN+MLPG is significantly better than HMM+MLPG. The DNN+DNN approach performs as well as HMM+MLPG while HMM+DNN is worse. In another experiment, they introduce a DNN before and after MLPG. While both approaches yield a MOS similar to DNN+MLPG, neither are statistically

[1] Note that since the two networks are consecutive in the framework, they can be considered a single network.

significantly better. make this part nicer to read or maybe just remove it? Quid of the MOS? Mention DMDN?

(Fan et al., 2014) supports that RNNs make natural acoustic models as they are able to learn a compact representation of complex and long-span functions, better so than shallow architectures such as the decision trees used in HMM-based TTS. Furthermore, the internal state of RNNs makes the mapping from an input frame to output acoustic features no longer deterministic, allowing for more variety in the synthesized audio. As RNNs are fit to generate temporally consistent series, the static features can directly be determined by the acoustic model, alleviating the need for dynamic features and MLPG. The authors present two RNNs: Hybrid_A with three feed-forward layers followed by one bidirectional (BDLSTM) layer and Hybrid_B with two feed-forward layers followed by two BDLSTM layers. They argue that deeper structures of BDLSTM would worsen the performance due to imprecise gradient computation. They compare these networks against the HMM and DNN based approaches described previously, using objective and subjective measures. The objective measures are compared with the ground truth: log spectral distance (LSD), voiced/unvoiced (V/UV) error rate and fundamental frequency (F0) distortion in root mean squared error (RMSE). The subjective measure is a preference test where participants must choose between two audio samples of different models and have the option to select neither. Results are shown in figures 3 and 4. DNN_A is 6 layers deep with 512 units per layer while DNN_B is 3 layers deep with 1024 units per layer. These two networks perform very similarly. Hybrid_B systematically performs better than the other approaches.



Measures Model	LSD (dB)	V/U Error rate	F0 RMSE (Hz)
HMM (2.89M)	3.74	5.8%	17.7
DNN_A (1. 55M)	3.73	5.8%	15.8
DNN_B (2.59 M)	3.73	5.9%	15.9
Hybrid_A (2.30M)	3.61	5.7%	16.4
Hybrid_B (3.61M)	3.54	5.6%	15.8

Figure 3: Performance of the different frameworks evaluated on objective measures. In parentheses are the number of parameters of each acoustic model.

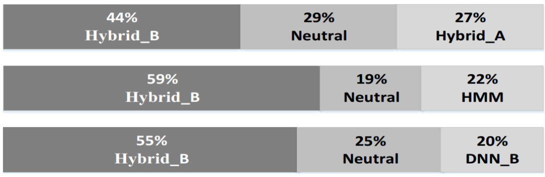


Figure 4: Two by two comparisons of some of the frameworks in terms of preferences.

Later, a substantial breakthrough in TTS is achieved with the coming of Wavenet (van den Oord et al., 2016). Wavenet is a deep convolutional neural network that, for a raw audio waveform, models the distribution of a single sample given all previous ones. It is thus possible to generate audio by predicting samples one at a time in an autoregressive fashion. Wavenet leverages stacks of one-dimensional dilated convolutions with a dilation factor increasing exponentially with the layer depth, which allows for a very large receptive field. The output is a categorical distribution (using a softmax layer) over sample values. For tractability reasons, the output signal is restricted to 256 values that correspond to a μ-law quantization, which is invertible. The MOS resulting from one of the experiments suggests that there is no statistically significant degradation in the naturalness of speech quantized in this manner. The architecture of Wavenet comes with several non-trivial components, described in later sections of this document link to the section. On its own, a trained Wavenet generates sound alike the training data but without semantics. The network must be locally conditioned on linguistic contexts to achieve TTS synthesis. Furthermore, it allows for conditioning with respect to a vector constant at every timestep (global conditioning), which can be used to designate the speaker identity. The authors suggest that Wavenet is able to encode the embedding of several speakers seen in training with a shared internal representation. Speaker identities are given as a one-hot encoding. The audio generated by Wavenet requires no post-processing other than the inversion of the μ-law. TTS with Wavenet does not exactly fit as an SPSS system considering that it does not produce clear intermediate acoustic features and instead serves as both a statistical model and a vocoder in a single pipeline. Performance scores are reported in Figure 5. The parametric approach is an LSTM-based system while the other is an HMM-driven unit selection concatenative system (not detailed in this document). Notice how the results vary between US English and Mandarin Chinese, showing that TTS performance is not language agnostic. recheck all the info on wavenet later + F0??

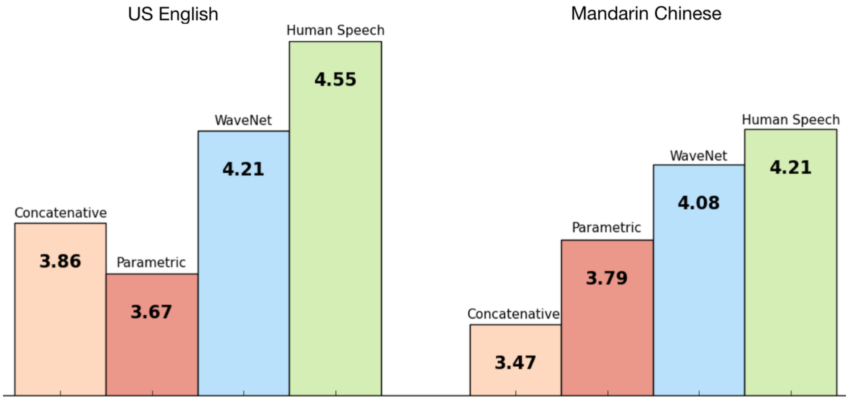


Figure 5: MOS of Wavenet's performance compared with a parametric and concatenative approach as well as with natural speech.

Deep Voice Arik et al. (2017) proposes a fully neural TTS framework that exploits Wavenet among other deep architectures. Deep Voice stands out by making use of only a few intermediate features: phonemes with stress annotations, phoneme durations, and F0. This has the advantage of making the framework easily transferable to new domains with little engineering effort as complex linguistic features need not be derived. The neural networks intervening in Deep Voice are: a grapheme-to-phoneme model that converts text to phonemes, a segmentation model that aligns a sequence of phonemes to an audio segment, a phoneme duration model that predicts the duration of each phoneme at generation time, a fundamental frequency model that predicts the V/UV flag with F0 values for voiced parts and finally, Wavenet which acts as an audio synthesis model. In all the works we presented previously, only the audio synthesis model was not manually engineered. While others have researched the use of neural networks for some of these components, Deep Voice is the first to simultaneously employ them all in a single framework. However, the grapheme-to-phoneme model is only used as a fallback for words not present in a phoneme dictionary. The roles and interactions of these components at training and inference time are shown in figure 6. Deep Voice also improves on the inference speed of Wavenet, with a speedup of up to 400. This allows for real-time or near real-time execution, with a tunable speed/quality trade-off. Deep Voice does not yield state of the art results but instead serves as groundwork for future researches. discuss results .

Skipping the following, in order:

Tacotron, Deep voice 2, Deep voice 3

I'll see later whether or not I should read and include them

Tacotron 2

Few samples

SV2TTS

Extensions?

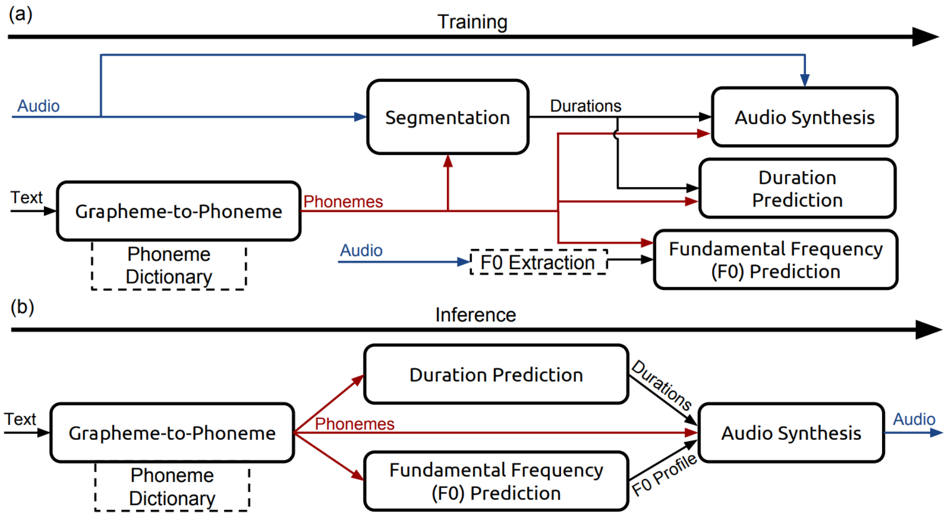


Figure 6: The training (a) and inference (b) procedure of Deep Voice. On the left of the image are the inputs, on the right are the outputs. Note that the segmentation model is only used for training.

References

Mary E. Beckman and Gayle Ayers Elam. Guidelines for tobi labelling, 03 1997.

Yoshihiko; Toda Tomoki; Zen Heiga; Yamagishi Junichi; Oura Keiichiro Tokuda, Keiichi; Nankaku. Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, 101, 05 2013. doi: 10.1109/JPROC.2013.2251852.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *EUROSPEECH*, 1999.

K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 3, pages 1315–1318 vol.3, June 2000. doi: 10.1109/ICASSP.2000.861820.

S. Imai. Cepstral analysis synthesis on the mel frequency scale. In *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 93–96, April 1983. doi: 10.1109/ICASSP.1983.1172250.

Takayoshi Yoshimura, Takashi Masuko, Keiichi Tokuda, Takao Kobayashi, and Tadashi Kitamura. Speaker interpolation in hmm-based speech synthesis system. In *EUROSPEECH*, 1997.

H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7962–7966, May 2013. doi: 10.1109/ICASSP.2013.6639215.

Y. Qian, Y. Fan, W. Hu, and F. K. Soong. On the training aspects of deep neural network (dnn) for parametric tts synthesis. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3829–3833, May 2014. doi: 10.1109/ICASSP.2014.6854318.

K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. The effect of neural networks in statistical parametric speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4455–4459, April 2015. doi: 10.1109/ICASSP.2015.7178813.

Y Fan, Yuang Qian, Feng-Long Xie, and Frank Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. pages 1964–1968, 01 2014.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL `http://arxiv.org/abs/1609.03499`.

Sercan Ömer Arik, Mike Chrzanowski, Adam Coates, Greg Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825, 2017. URL `http://arxiv.org/abs/1702.07825`.