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 TSS-specific words that may be worth explaining: coarticulation, linguistic context, spectral envelope, fundamental frequency, supra-segmental

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

Concise presentation of the problem

2.1 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 [BE97] is often used.

Talk about evaluation metrics (mainly MOS)?

2.2 State of the art in multispeaker TTS

Previous state of the art in SPSS includes hidden Markov models (HMM) based speech synthesis [Tok13]. 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 be partitioned over the different contexts, which is very data inefficient. Instead, contexts are clustered using decision trees and an HMM is learned for each cluster [YTM+99].

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 [TYM+00]. These static features are then fed through the MLSA [Ima83] vocoder. It is possible to modify the voice generated by conditioning on a speaker or tuning the generated speech parameters with adaptation or interpolation techniques (e.g. [YMT+97] elaborate a bit on these techniques?), making HMM-based speech synthesis a multispeaker TTS system. Compare with concatenative see [ZSS13] and ieeexplore.ieee.org/document/541110.

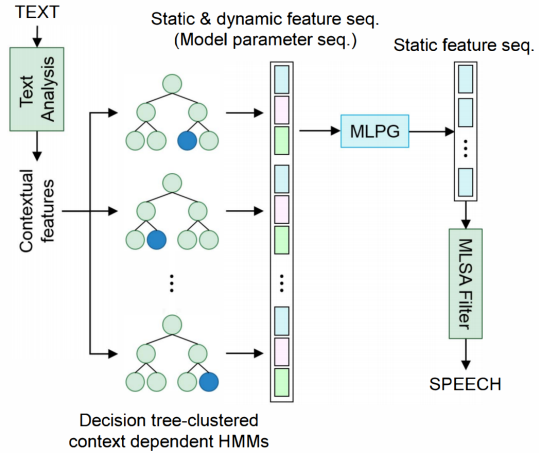

Figure 1: The general HMM-based TTS synthesis approach.

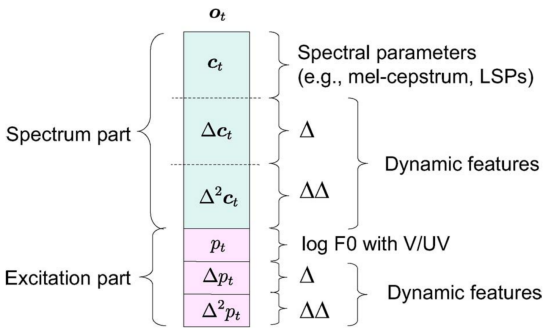

Figure 2: Dynamic and static features produced by the HMMs. F0 is the fundamental frequency and V/UV is the voicing flag.

Improvements to this framework were later brought by feed-forward deep neural networks (DNN), as a result of progress in both hardware and software. [ZSS13] 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 [QFHS14]. [HONT15] 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 [Tok13] and [ZSS13]. 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 significantly better. make this part nicer to read. Quid of the MOS?

[FQXS14] 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

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

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 true ground: log spectral distance (LSD), V/UV error rate and F0 distortion in root mean squared error (RMSE). The subjective measure is a preference test where participants 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.

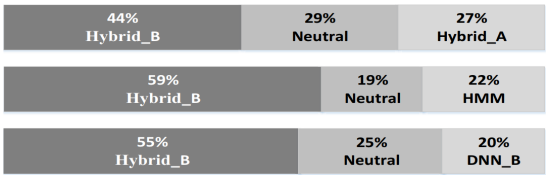

Figure 4: Two by two comparisons of some of the frameworks in terms of preferences.

Wavenet: breakthrough in TTS with raw waveform gen
Take images from https://deepmind.com/blog/wavenet-generative-model-raw-audio/ ?
Dilated causal convolutions
Condition on a speaker identity
Tacotron
Deep voice (1, 2, 3 + few samples), Tacotron 2
SV2TTS
Extensions?

References

[BE97] Mary E. Beckman and Gayle Ayers Elam. Guidelines for tobi labelling, 03 1997.

[FQXS14] Y Fan, Yuang Qian, Feng-Long Xie, and Frank Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. pages 1964–1968, 01 2014.

[HONT15] 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.

[Ima83] 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.

[QFHS14] 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.

[Tok13] 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.

[TYM+00] 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.

[YMT+97] Takayoshi Yoshimura, Takashi Masuko, Keiichi Tokuda, Takao Kobayashi, and Tadashi Kitamura. Speaker interpolation in hmm-based speech synthesis system. In EUROSPEECH, 1997.

[YTM+99] 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.

[ZSS13] 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.