Introducing Prosodic Speaker Identity for a Better Expressive Speech Synthesis Control

Aghilas Sini, Sébastien Le Maguer, Damien Lolive, Elisabeth Delais-Roussarie

10th International Conference on Speech Prosody 2020
25-28 May 2020, Tokyo, Japan

Introducing Prosodic Speaker Identity for a Better Expressive Speech Synthesis Control

Aghilas Sini[1], *Sébastien Le Maguer*[2], *Damien Lolive*[1], *Elisabeth Delais-Roussarie*[3]

[1]Univ. Rennes, IRISA, Lannion, France
[2]Sigmedia Lab, ADAPT Centre, EE Engineering, Trinity College Dublin, Dublin, Ireland
[3]Université de Nantes, UMR 6310 - LLLING, Nantes
{aghilas.sini,damien.lolive}@irisa.fr, lemagues@tcd.ie,
elisabeth.delais-roussarie@univ-nantes.fr

Abstract

To have more control over Text-to-Speech (TTS) synthesis and to improve expressivity, it is necessary to disentangle prosodic information carried by the speaker's voice identity from the one belonging to linguistic properties. In this paper, we propose to analyze how information related to speaker voice identity affects a Deep Neural Network (DNN) based multi-speaker speech synthesis model. To do so, we feed the network with a vector encoding speaker information in addition to a set of basic linguistic features. We then compare three main speaker coding configurations: *a*) simple one-hot vector describing the speaker gender and identifier ; *b*) an embedding vector extracted from a speaker recognition pre-trained model ; *c*) a prosodic vector which summarizes information such as melody, intensity, and duration. To measure the impact of the input feature vector, we investigate the representation of the latent space at the output of the first layer of the network. The aim is to have an overview of our data representation and model behavior. Furthermore, we conducted a subjective assessment to validate the result. Results show that the prosodic identity of the speaker is captured by the model and therefore allows the user to control more precisely synthesis.

Index Terms: Multi-speaker TTS, Speaker control, Expressive TTS, Deep Learning

1. Introduction

The quality of speech synthesis systems has drastically increased during the last years. Thanks to the deep learning paradigm, it is now possible to generate speech, which sounds almost like human speech. However, the control over models remains challenging because of their complexity.

Expressive speech synthesis relies on adequate control on the prosodic parameters. These parameters depend on the linguistic features of the text to read as well as information related to the voice used for synthesizing the speech.

Therefore, disentangling the speaker characteristics from the linguistic content is a key feature to control the rendering of the synthesis.

Disentangling the speaker characteristics from the linguistic content is even more crucial to have proper control in multi-speaker Statistical Parametric Speech Synthesis (SPSS) as, by definition, the model should produce a speech signal corresponding to one consistent speaker. Counting on the robustness of multi-speaker modelling, studies show that expressive speech synthesis systems can benefit from such an environment [1], although it raises other challenges related to recording conditions [2], speaker coding [3] and controllability [4, 5, 6, 7].

Therefore, in this paper, we propose to investigate if using a naive but fully controllable representation of the prosody, the model can separate the speaker characteristics from the linguistic features in a standard DNN TTS multi-speaker environment.

This article is structured as follows. The different speaker coding configurations are presented in Section 2. Section 3 gives an overview of the methodology and Section 4 details the experiments we conducted to analyze the influence of these configurations on the model. Finally, in Section 5, we go through the results of the experiments using complementary objective analysis methodologies and subjective assessment.

2. Speaker Coding

To encode the speaker voice characteristics, we are using three different configurations from the most opaque (OneHot-Vector) to the most controllable one (P-Vector). The intermediate representation (X-Vector) has been added as it is a state of the art representation for the speaker identification domain.

2.1. OneHot-Vector (OHV)

This configuration to encode the speaker information for DNN based speech synthesis has been explored in [3]. As a first and intuitive choice for speaker encoding, we propose a simple one-hot vector of two parts: (1) gender (female/male) as this is the highest level distinction we can do, (2) identifier of the speaker to distinguish speakers intra-gender. This approach is the one making the control of the synthesis the most complicated as we just have a discrete choice. Thus it does not take into account the acoustic proximity between speakers.

2.2. X-Vector

X-Vectors [8] are the state of the art represention used in the speaker identification field. To get the X-Vectors, we extract embedded vectors independently on the text using a pre-trained model[1]. As stated before, this model was initially trained for a speaker verification task [8, 9, 10] using NIST SRE recipe supported in the Kaldi toolkit. The details about the recipe and the pretained model are available in author's github[2]. This configuration is more detailed than the OneHot-Vector but still remains difficult to control as the dimensions of the X-Vectors are difficult to interpret.

2.3. P-Vector

The last configuration is the one we are proposing: the P-Vector. To characterize the speaker style and specificity of an expressive

[1]https://kaldi-asr.org/models/m3
[2]https://david-ryan-snyder.github.io/2017/10/04/model_sre16_v2.html

10.21437/SpeechProsody.2020-191

voice, we propose to use the breath group as the functional unit to build a vector able to cover high-level prosodic information which are difficult to predict from the text. A P-Vector is defined by the following features:

- F0-range: for each vowel of the breath group, we are computing the median values. Then, considering $F0_{min}$ and $F0_{max}$, respectively, the minimum and the maximum median values, we computed the scaled fundamental frequency (F_0) range the following way:

$$\begin{cases} F_{0min} & = & \min(V^0_{F_{0median}}, \ldots, V^M_{F_{0median}}) \\ F_{0max} & = & \max(V^0_{F_{0median}}, \ldots, V^M_{F_{0median}}) \\ F_{0range} & = & 12 \times log_2(\frac{F_{0min}}{F_{0max}}) \end{cases}$$

  where M represents the number of vowels within the breath group and $V^i_{F_{0median}}$ stands for the median F_0 value of the ith vowel within the breath group;
- Melodic pattern: for each vowel contained in a given breath group, the $V_{F_{0median}}$ has been extracted. The resulting sequence of values has been interpolated using a cubic spline. Then, a set of five equidistant values (at each 20% of the breath group duration starting from 10%) has been selected.
- Energy pattern: the same computation as the previous one is done on $V_{logEnergy}$.
- Articulation Rate: it is the number of syllables per seconds computed at the breath group level ignoring silences;
- Duration of breath group in second;
- Duration of pauses around the breath group in second.

This way, we obtain a fully controllable feature vector whose dimensions can be adequately interpreted.

3. Analysis Methodology

The experiments and analyses presented in this work were carried out within the Merlin [11] framework. We used the default configuration proposed in the toolkit, then we integrated the speaker coding vectors to achieve a multi-speaker TTS model.

3.1. Input and Output features

The input feature vector can be viewed as two concatenated vectors corresponding to two parts: a linguistic part and a speaker coding part. The first 319 coefficients correspond to the linguistic description of the utterance. This part is based on the standard feature set for English described in [12] that we have adapted for French. The main differences with the English feature set concerns the accentuation. Indeed, as the accentuation information in French is strongly correlated to the Part of Speech (POS) information, we therefore consider that the POS information, already present in the vector, is enough to encode the accentuation information. The coefficients from dimension 320 and beyond are the speaker code. The size of this part varies according to the configurations under study (e.g. OneHot-Vector, X-Vector or P-Vector).

The output feature vector contains the standard coefficient vector composed by the Voiced/Unvoiced (VUV) flag, the $\log F_0$, the mel-generalized cepstrum (MGC), the Band Aperiodicity Parameter (BAP), and their dynamic counterparts. This leads to a vector of 265 coefficients.

Finally, the input and output vectors are normalized using, respectively, Min-Max Normalization (MMN) and Mean Variance Normalization (MVN) methods.

3.2. Method

The main goal of the paper is to see if and how the content of the input vector influences the ability to separate speaker-related information in a DNN-based TTS system. To do so, we train several systems differing by the structure of the input vectors provided. Once the different systems are learned, to analyze if the various configurations are guiding the models to capture speaker specificities, we propose to measure differences at the output of the first hidden layer as well as at the output of the model. Two types of analyses are then done:

- Standard objective measures: mel-cepstral distortion (MCD), BAP distortion, F_0 Root Mean Square Error (RMSE), F_0 correlation, VUV error rate, RMSE on the duration and duration correlation;
- A visual analysis protocol: a Principal Component Analysis (PCA) on the first hidden layer output is computed. Then, we visualize the main dimensions and analyze the results in function of the speakers to see if speaker-dependent information is captured by the model. We perform PCA at the end of each epoch on the validation dataset. We choose to do the analysis at this stage of the network because it is easier to interpret and quantify the variation brought by the input.

We also compare different epochs to see how the models are evolving. This monitoring is interesting since it enables to check quickly if the structure of the input vectors has an impact on speaker separability.

4. Experimental setup

4.1. Dataset

A corpus of multiple speakers was collected from two different libraries: LitteratureAudio[3] and LibriVox[4].

This database contains fictional french audiobooks published between the 18th and the 20th century. All the transcriptions of the corpus are freely available in wikisource[5]. The text is split into paragraphs and then force-aligned to corresponding speech using JTrans[13]. The speech signals are sampled at 48 kHz. All the meta-data information related to describe the book (speaker identifier, library name, ...) was removed. From the designed corpus, two groups of data were defined:

- *parallel data*: this group contains 5 audiobooks; each transcription was read by at least 2 speakers. In total, the data for 9 speakers has been collected including 4 females. Voices were selected by an informal listening test considering their recording conditions (non-audible difference), and the fact that the voice quality of the speakers are quite different.
- *non-parallel data*: for each speaker in the parallel data, 1h of extra speech has been collected with no overlap in the transcription. This set of data is used to evaluate robustness and performance of the speaker encoder input.

The procedure used to achieve the annotation process and to extract the linguistic features is described in [14].

4.2. Models configuration

To achieve training and synthesis, we used the Merlin toolkit [11]. The architecture of the model is a Feed-Forward

[3] http://www.litteratureaudio.com/
[4] https://librivox.org/
[5] https://fr.wikisource.org/wiki/Wikisource:Accueil

Table 1: *Objective results for multi-speaker modeling, considering five speaker code configurations. mel-cepstral distortion (MCD), Band Aperiodicity Parameter (BAP), Root Mean Square Error (RMSE), Voiced/Unvoiced (VUV) and Correlation (CORR) between the predicted and the original coefficients. For the F_0, RMSE and CORR are computed exclusively on voiced frames.*

OHV	X-Vector	P-Vector	MCD (dB)	BAP (dB)	F0		VUV	Duration	
					RMSE (Hz)	CORR		RMSE (ms)	CORR
✓			5.833	0.301	32.597	0.807	8.950	9.232	0.558
	✓		5.935	0.303	33.018	0.801	8.971	8.889	0.601
		✓	5.748	0.296	32.203	0.811	8.851	8.883	0.604
✓		✓	5.756	0.297	32.169	0.810	8.944	8.860	0.607
	✓	✓	5.755	0.297	32.043	0.812	8.915	8.836	0.609

DNN (FF-DNN) with 4 hidden layers. During the experiments, we changed first layer size to be 128, 256 or 512 neurons without any significant change. The last three layers have a fixed number of 512 neurons. The hidden layers use the *tanh* activation function and the output layer uses a linear activation function. We applied batch-training paradigm with a batch size of 256. The maximum number of epochs is set to 25 including 10 warm-up epochs. The learning rate is initially set to 0.002 for warm-up epochs and after that reduced by 50% for each epoch. Similarly, the momentum is set to 0.3 for warm-up epochs and to 0.9 otherwise. Finally, we used L2-regularization with a weight set to 10^{-5}. Models are trained considering speaker coding schemes with the following dimensions: 2 for OHV, 32 for X-Vector and 9 for P-Vector.

5. Results

5.1. Standard measurements

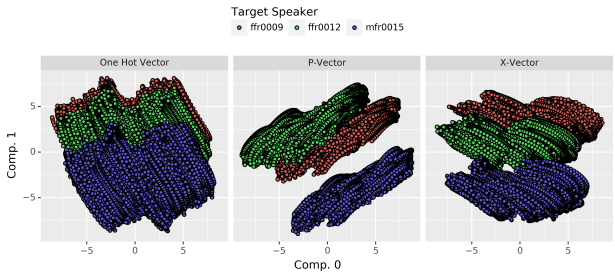
In order to evaluate DNN-based TTS synthesis, the proposed method was applied to train models on the parallel training set and then on the non-parallel training set.

All the models were evaluated using MCD, BAP distortion, RMSE on F_0 and duration, VUV rate and Correlation (CORR) on F_0 and duration, between the predicted and the original coefficients. In this paper, only the objective results concerning the non-parallel training dataset are reported as similar results have been observed on the parallel training dataset.

As shown in Table 1, the systems involving the P-Vector beat the baseline system in all kinds of objective measures.

5.2. Visualizing the first hidden-layer output

Figure 1: *PCA projection for the parallel data during the validation phase, the speaker identity is encoded as following (F/M: Female/Male, FR: French, ID:XXXX).*



PCA[6] has been applied on the output of the first hidden layer to reduce the number of dimensions down to the two main ones.

[6]We choose PCA to find out the independent variables that hold the speaker's identity.

Figure 2: *PCA projection for the non parallel data during the validation phase.*

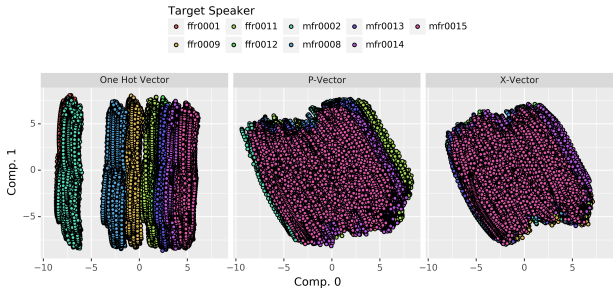


Figure 1 and Figure 2 illustrate respectively the parallel data and non-parallel data projections for the different configurations. With non-parallel data, both X-Vector and P-Vector do not show a clear separation between speakers compared to OneHot-Vector. However, using parallel data, P-Vector and X-Vector manage to discriminate the speakers.

The first explanation for this behavior is that with non-parallel data, the linguistic, prosodic and phonetic context variability are dominant and most of the variation is hold by those components. As the data are non parallel, the neural network has more difficulty to distinguish the speakers. The second possible explanation is that the size and complexity of X-Vector and P-Vector bring more sparsity in the latent space, which is not the case with OneHot-Vector. Finally, it seems that X-Vector and P-Vector can be used equally to bring speaker control to the system but due to the lower complexity of P-Vector, this representation might be preferable.

The visualization of the evolution of the latent space projection at different epochs is illustrated on Figure 3. It enables to monitor the learning process and check quickly the impact of the vector structure on the speaker separation. Here, we can notice that from epoch 10, the projected latent space is quite stable and the speakers well separated.

5.3. Subjective Evaluation

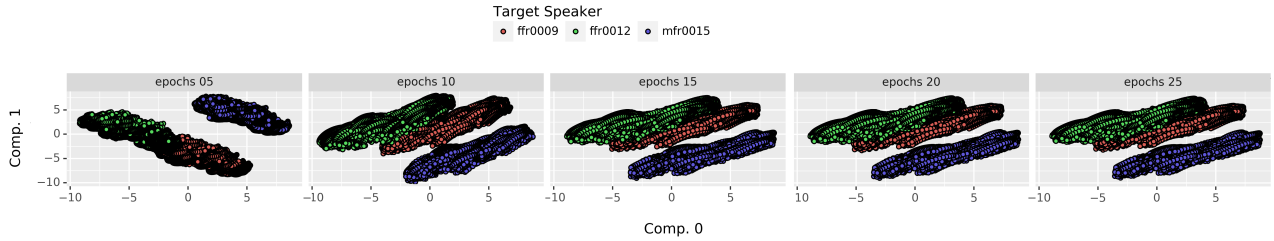
5.3.1. Evaluation protocol

In order to validate our proposition, we conducted a subjective evaluation based on the MUSHRA protocol [15]. The reference is the re-synthesis using the **world** vocoder. We use a speaker dependent baseline (**spkdep**) as well as a speaker independent model (**spkadapt**)[7]. Then, we evaluated the isolated configurations (**OneHot-Vector**, **X-Vector** and **P-Vector**).

The duration of each of the 54 samples presented to the listeners varied in a from 4s to 6s. The ratio of speech breaks

[7]https://github.com/AghilasSini/merlin/tree/master/egs/speaker_adaptation

Figure 3: *Visualization of the latent representation in case of P-Vector using parallel data. We can notice the separation of the speakers representation from epoch 5 to epoch 25.*



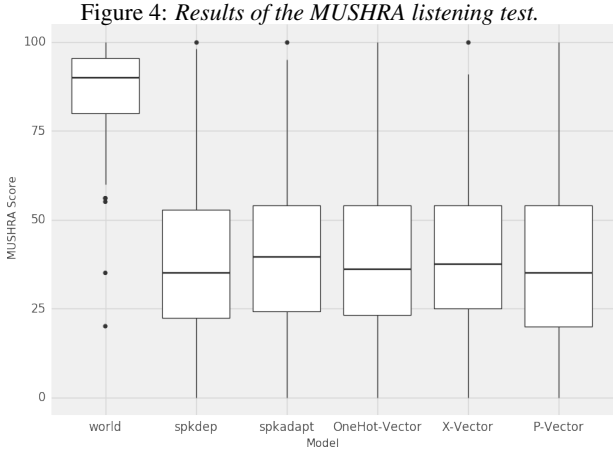
present in the selected samples does not exceed the quarter of the total duration of the sample.

One evaluation instance is composed by 9 steps including all the models presented before. 30 listeners completed the evaluation. They were French native speakers aged between 24 and 45. The majority of them have experience with listening tests but are not necessarily experts in the annotation of audio files. All materials are available in the dedicated repository[8].

5.3.2. *Discussion and results*

The results of the evaluation are presented in Figure 4. From them, we can see that the reference is correctly identified, which guarantees the validity of the evaluation. It seems that some annotators estimate that even the reference was not good enough for some samples which explains the fact that the reference did not achieve a score of 100. Then, considering the models evaluated, no system is outperforming the other ones. This leads us to conclude that listeners do not distinguish major differences. To verify that the listeners didn't perceive minor differences,

Figure 4: *Results of the MUSHRA listening test.*

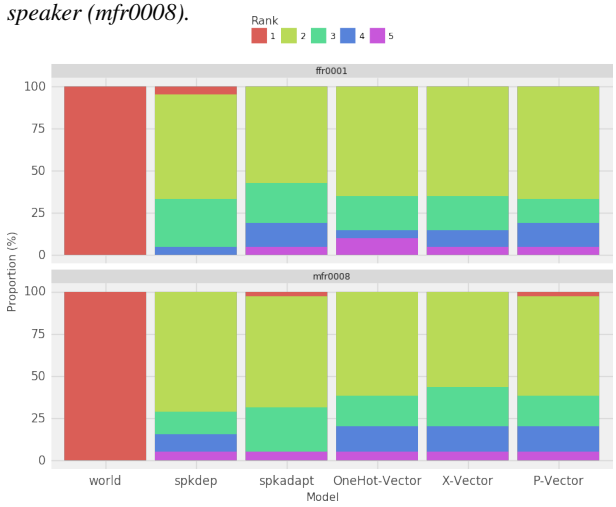


we also computed the rank of each system for each step based on its score. Figure 5 presents the results of two speakers, but the results for the other speakers of the corpus are following the same patterns. Results of two speakers are presented in Figure 5. In all the cases, the best configuration is the reference. When we compare the results of the synthetic systems, we can see that their proportions are globally similar. Finally, the second rank is also predominant for each system, expressing that listeners have graded multiple systems as equivalent.

Based on all these results, we can conclude that the quality produced by all the synthesis systems are equivalent.

[8]https://github.com/AghilasSini/SpeechProsody2020

Figure 5: *Ranking score of a female speaker (ffr001) and a male speaker (mfr0008).*



6. Conclusion

In this paper, we have evaluated different speaker coding schemes both objectively and subjectively in a DNN-based framework. All the evaluations conducted show no difference in the quality of the modeling of the three different speaker coding schemes. These results are valid in both studied cases, parallel or non-parallel data for multi-speaker modeling. Moreover, the speaker coding scheme we proposed, the P-Vector, provides better control of the modeling. This investigation confirms the relevance of the prosodic parameters that we choose to build the prosodic identity of speakers. However, a close look at this representation shows that the intra-speaker prosodic variation related to discourse changes (narration, dialog) is excluded.

These results are encouraging and suggest further research work. Furthermore, we plan the evaluation of the robustness of the proposed speaker coding on a dataset that contains more speakers and investigating other factors such as language, literary genre, discourse typography, and structure.

7. Acknowledgements

This research was conducted under the ANR (French National Research Agency) project SynPaFlex ANR-15-CE23-0015 and the LABEX EFL (Empirical Foundations in Linguistics) ANR-10-LABEX-0083. It has been also conducted with the financial support of IRC under Grant Agreement No. 208222/15425 at the ADAPT SFI Research Centre at Trinity College Dublin. The ADAPT SFI Centre for Digital Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

8. References

[1] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4475–4479.

[2] W. Hsu, Y. Zhang, R. J. Weiss, Y. Chung, Y. Wang, Y. Wu, and J. Glass, "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5901–5905.

[3] N. Hojo, Y. Ijima, and H. Mizuno, "Dnn-based speech synthesis using speaker codes," *IEICE TRANSACTIONS on Information and Systems*, vol. 101, no. 2, pp. 462–472, 2018.

[4] G. E. Henter, X. Wang, and J. Yamagishi, "Deep encoder-decoder models for unsupervised learning of controllable speech synthesis," *arXiv preprint arXiv:1807.11470*, 2018.

[5] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[6] A. Lazaridis, B. Potard, and P. N. Garner, "Dnn-based speech synthesis: Importance of input features and training data," in *International Conference on Speech and Computer*. Springer, 2015, pp. 193–200.

[7] Y. Bian, C. Chen, Y. Kang, and Z. Pan, "Multi-reference tacotron by intercross training for style disentangling, transfer and control in speech synthesis," *arXiv preprint arXiv:1904.02373*, 2019.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.

[10] L. Xu, R. K. Das, E. Yılmaz, J. Yang, and H. Li, "Generative x-vectors for text-independent speaker verification," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1014–1020.

[11] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system." in *SSW*, 2016, pp. 202–207.

[12] K. Tokuda, H. Zen, and A. W. Black, "An hmm-based speech synthesis system applied to english," in *IEEE Speech Synthesis Workshop*, 2002, pp. 227–230.

[13] C. Cerisara, O. Mella, and D. Fohr, "Jtrans: an open-source software for semi-automatic text-to-speech alignment," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[14] A. SINI, D. Lolive, G. Vidal, M. Tahon, and É. Delais-Roussarie, "SynPaFlex-corpus: An expressive French audiobooks corpus dedicated to expressive speech synthesis." in *Proceedings of the 11th Language Resources and Evaluation Conference*. Miyazaki, Japan: European Language Resource Association, May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1677

[15] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.