SPEECH/NON-SPEECH CLASSIFICATION SLIGHTLY IMPROVES SYNTHESIS QUALITY IN PAULE

Konstantin Sering

Eberhard Karls Universität Tübingen
konstantin.sering@uni-tuebingen.de

Abstract: One of the tasks PAULE[1, 2] solves is finding suitable control parameter (cp-)trajectories for a given target acoustic. These cp-trajectories can be used to synthesize speech with the articulatory speech synthesizer of the VocalTractLab (VTL) [3]. If the target acoustic contains substantial microphone noise or other background noises, occasionally PAULE optimizes not for the speech in the target, but for this background noises. By adding a speech/non-speech classifier to the feedback and planning-loop in PAULE this resynthesis of background noises should be mitigated. Unfortunately, the improvements were minor, which might be due to uninformative gradients of the classifier. The importance of informative gradients and the use classifiers to adapt PAULE to different tasks are explained and discussed.

1 Introduction

The Predictive Articulatory speech synthesis model Utilizing Lexical Embeddings (PAULE)[1] [1, 2] is a control model for the articulatory speech synthesizer in VocalTractLab (VTL) [3, 4]. PAULE can solve two inverse problems: First, PAULE can be used to generate articulatory control parameter (cp-)trajectories for a given semantic embedding vector from a set of 4,311 German word types and a production duration; second, PAULE can be used to do copy-synthesis on any audio recording. In this copy-synthesis or acoustic-only inverse problem, which is the focus of this work, PAULE finds suitable cp-trajectories for a given target acoustic recording. These cp-trajectories serve as inputs to the VTL synthesizer and lead to a speech synthesis that is acoustically and semantically close to the target acoustic. The challenge arises from the high dimensionality and non-linearity of the movement trajectories of the tongue and jaw and the time series of the flapping frequency of the vocal folds (f0) and lung pressure together with all the other parameters needed by the VTL synthesizer to synthesize speech. The number of parameters (channels) in the VTL is 30. Together with a sampling rate of 401 Hz this leads for one second of speech to 12,030 parameter values. These 12,030 parameter values need to be inferred by PAULE. These control parameters (cps) are constrained by the geometry of the vocal tract and should fulfil certain external constraint. For example, there should not be any very rapid changes over time in the cps.

Up to now, PAULE is trained and evaluated on relatively clean and noise-free audio recordings. Giving PAULE an audio recording as target acoustic containing higher amounts of microphone noise leads to cp-trajectories, which try to not only emulate the containing target speech, but additionally or sometimes exclusively the microphone noise or present background sounds. As long as PAULE should not be used to do any beat boxing, this behaviour of the PAULE model is not desired. One mitigation, which was mainly implemented by us for now,

[1] `https://github.com/quantling/paule`

DOI: 10.35096/othr/pub-7095, URN: urn:nbn:de:bvb:898-opus4-70955

is to denoise the target signal and remove any background sounds as far as possible. With this contribution we want to focus on a different way of mitigating the problem, namely by adding a speech/non-speech classifier to the predictive planning loop in PAULE.

1.1 Motor space vs. Goal space

In order to understand why adding a classifier to the planning loop should help mitigating the problem of mimicking a noisy target acoustic, we first need to understand how the predictive principle is implemented in PAULE and how PAULE uses a goal space optimization procedure instead of a source space optimization. The source space for PAULE is the space of all possible cp-trajectories, which can be seen as a motor space.

To solve the inverse problem of copy-synthesis we need to find suitable cp-trajectories, which define the motor space. One possibility is using the target acoustic as inputs to a direct feed forward model that predicts cp-trajectories [5] or the use of a symbolic-segmentation approach, by first finding and aligning phonetic segments in the target acoustic and then using gestural-scores and a blending method to generate cp-trajectories [6]. Both of these approaches can be seen as motor space optimizations, which define a set of optimal motor trajectories and then use some continuation assumptions to drive the realised movements close to this optimal motor trajectories.

In contrast, the optimization scheme implemented by PAULE does not try to find any optimal cp-trajectories in the motor space. Instead it optimizes the predicted effect of the cp-trajectories in a goal space. So far PAULE implements two goal spaces, an acoustic goal space as a log-mel spectrogram with 60 Mel banks and a sampling rate of 200.5 Hz and a semantic embedding goal space of 300-dimensional fastText lexical embeddings vectors [7]. The quality of the cp-trajectories is not evaluated or driven by having an optimal reference cp-trajectories, but by minimizing the distance in the acoustic and semantic domain.

With the shift to an optimization in a goal space, we only need to understand and learn the distance structure within this goal space and then project this similarity structure into the motor space. Figure 1 shows a visual representation of this idea. Luckily, in the goal space of acoustics we can rely on substantial amounts of freely available speech audio recordings, which can be augmented by word labels. These speech audio recordings can be used to learn a similarity structure in the acoustic space. In the semantic goal space we can rely on the similarity structure of word embeddings, which usually rely on huge amounts of written text. The main assumption is, that if we have a close distance in the acoustic and semantic goal space, that the produced speech by the VTL is intelligible and to a human listener similar to the target acoustic in sound quality and meaning.

Furthermore, this goal space optimization renders the question "What is the optimal motor trajectory?" obsolete, because – in principle – many different cp-trajectories can be equally good and optimal. These cp-trajectories are – in principle – situational and therefore there is little gain in seeking for the "true" motor trajectory.

To help with the microphone noise in the target acoustic, we introduce a third goal space here, namely the speech/non-speech goal space. By classifying the acoustics in either speech-like acoustics or noise-like/ambient acoustics, we have a score on how speech-like any acoustics is and how distant this classification score is from a speech signal. This distance can subsequently be used to improve the cp-trajectories to make the resulting speech synthesis of the cp-trajectories more speech-like.

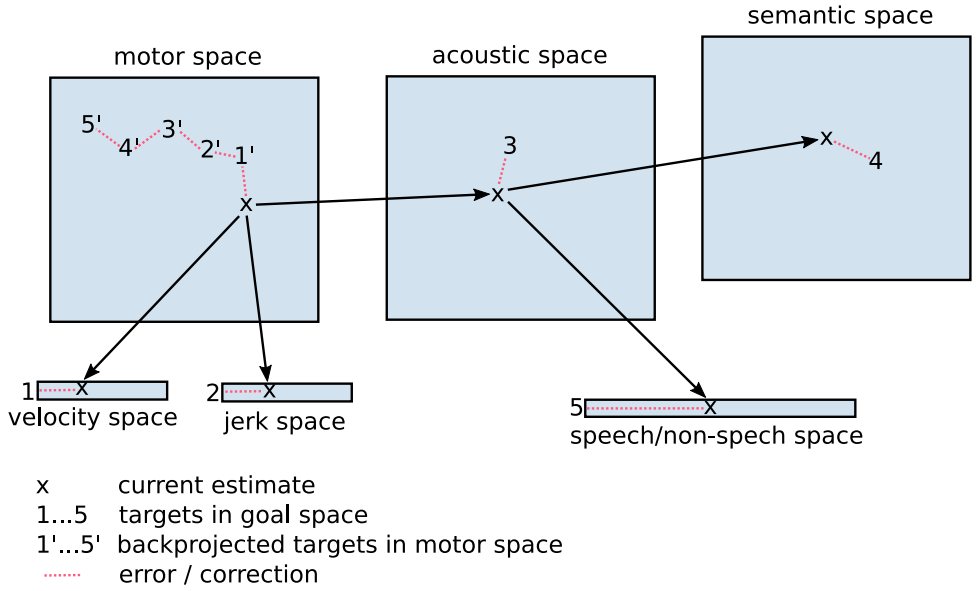

Figure 1 – The control-parameter (cp-)trajectories denoted as **x** in the motor space are projected to five goal spaces. In each goal space the projection is denoted by **x** and the target in each goal space has the numbers 1 to 5. As all the projections are gradient-aware the error, denoted as red dotted line, can be back propagated to the motor space. This allows for a correction of the original cp-trajectory **x** in the direction of the targets. As the correction is along the *local* gradients, it usually only gives an approximate correction. PAULE defaults to 24 correction steps in its planning loop. The motor space and the acoustic space for one second of speech in PAULE are 12,030-dimensional each. The semantic space is 300-dimensional and the velocity, jerk, and non-speech spaces are 1-dimensional.

1.2 Gradient

In order to use the speech-like score from the speech/non-speech classifier to improve the cp-trajectories the discrepancy between the speech-like score and the ideal result of being classified as 100% speech-like needs to be projected back into the motor space. I. e. we want to know how to change the cp-trajectories so that the acoustic synthesis gets more speech-like. Luckily, this back-projection is a standard method in the field of machine learning and artificial neural networks. Here it is called back-propagation and is usually used to update the internal weights of a neural network model. In the planning steps in PAULE, we use this backpropagated gradient information differently. We use the backpropagation to backproejct the error from the goal space into the motor space. This backpropagation is done along the gradients of the forward model. To backpropagate the speech/non-speech error, first the error gets backpropagated into the acoustic space and then further backwards to the motor space. The pathway is depicted in Figure 2 and the idea of backpropagating the error into the motor space is depicted in a simplified form in Figure 1. Using the gradients of the forward model to improve and plan the cp-trajectories is the key feature of the PAULE model and is described in detail in [2].

During the planning four different types of errors are minimized in PAULE: The first type of error is the one from the semantic goal space. This is minimized by backpropagating the error along the gradients of the embedder and the forward model to the current cp-trajectories (see Figure 2 for reference). The second type of error is the one from the acoustic goal space. This is minimized by backpropagating the acoustic error along the gradients of the forward model. The third type of error is defined internally on the cp-trajectories, i. e. in the motor space. It minimizes the velocity and jerk (third time derivative) profile from the current cp-trajectories. Minimization of the velocity leads to stationary trajectories, as long as no other types of errors play a role, and minimizing the jerk leads to constant force or ballistic trajectories. The last and fourth type of error is the newly introduced one from the speech/non-speech classifier. This is minimized by backpropagating the speech/non-speech classification and the forward prediction

of the forward model. These four types of errors are jointly minimized by first creating an additive error through a weighted sum. The error is often called loss and denoted by L. In PAULE the planning loss L_{planning} is defined as

$$L_{\text{planning}} = L_{\text{semantic}} + \beta_1 \cdot L_{\text{acoustic}} + \beta_2 \cdot L_{\text{cp}} + \beta_3 \cdot L_{\text{speech-like}}.$$

The β_i are the weighting parameters and define, which type of error gets which importance. For the time being these β-values are given a fixed value, but adaptively setting these β-values could be done. Adaptive weighting schemes comes with additional challenges though, which is shortly discussed later.

Note that the newly introduced speech/non-speech goal space only needs to be implemented as a gradient aware classifier. Therefore, as long as you can compute some speech- or articulation-based measure, and this computation can be written in a gradient-aware manner it can be used like the speech/non-speech classifier. Examples could be emotional measures on the speech, but also classifiers for languages or speakers. Additionally, this can be used to implement soft constraints like closing the mouth, if nothing is said anymore.

2 Methods

The idea of training a speech/non-speech classifier to improve synthesis quality in PAULE is a two step process. First, the classifier is trained in a supervised manner. Second, the classifier is integrated into the planning loop of PAULE as depicted in Figure 2. The speech/non-speech classifier is trained on the acoustic representation of PAULE. The acoustic representation is a log-mel spectrogram with a frequency range of 10 Hz to 12,000 Hz. The frequency range is split into 60 bands and the time resolution is 200.5 Hz. The window size of the fft is 23.2 milliseconds. This log-mel spectrogram serves as the input to the classifier. The output is a single dimension, which encodes the likelihood of being non-speech like as speech is labeled with a zero and non-speech is labeled with a one.

2.1 Model architectures

As the classification problem is a sequence to fixed-value problem, an encoder only Transformer architecture [8] is used here. The input data is encoded with a positional encoding, then feed into a Transformer encoder layer with 60 model dimensions, 6 heads, and 1,024 feedforward dimensions. The activation function is a Gaussian Error Linear Unit (GELU) activation function. Three encoder layers are used. Of the outputs of the encoder layers the mean along the sequence dimension is computed and then linearly mapped to 20 dimensions. These 20 dimensional vector is squashed with another GELU function and then linearly mapped to the single output dimension. For the loss a binary cross-entropy loss is used.

2.2 Training data

Three data sources are used as training data. Human speech recordings from the German Common Voice corpus [9] and a segment-based resynthesis of these recordings [6] are used for the speech category and are therefore labeled as zero. The recordings of environmental sounds from the ESC-50 dataset [10] are used as non-speech. From the speech recordings single words are first spliced out of the full recording. From the ESC-50 dataset non-zero intervals of the same length are extracted randomly. For all sounds log-mel-spectrograms (lms) are calculated. This leads to 63,525 training samples and 15,288 validation samples. The shortest log-mel spectrogram has a sequence length of 18 and the longest has a sequence length of 472.

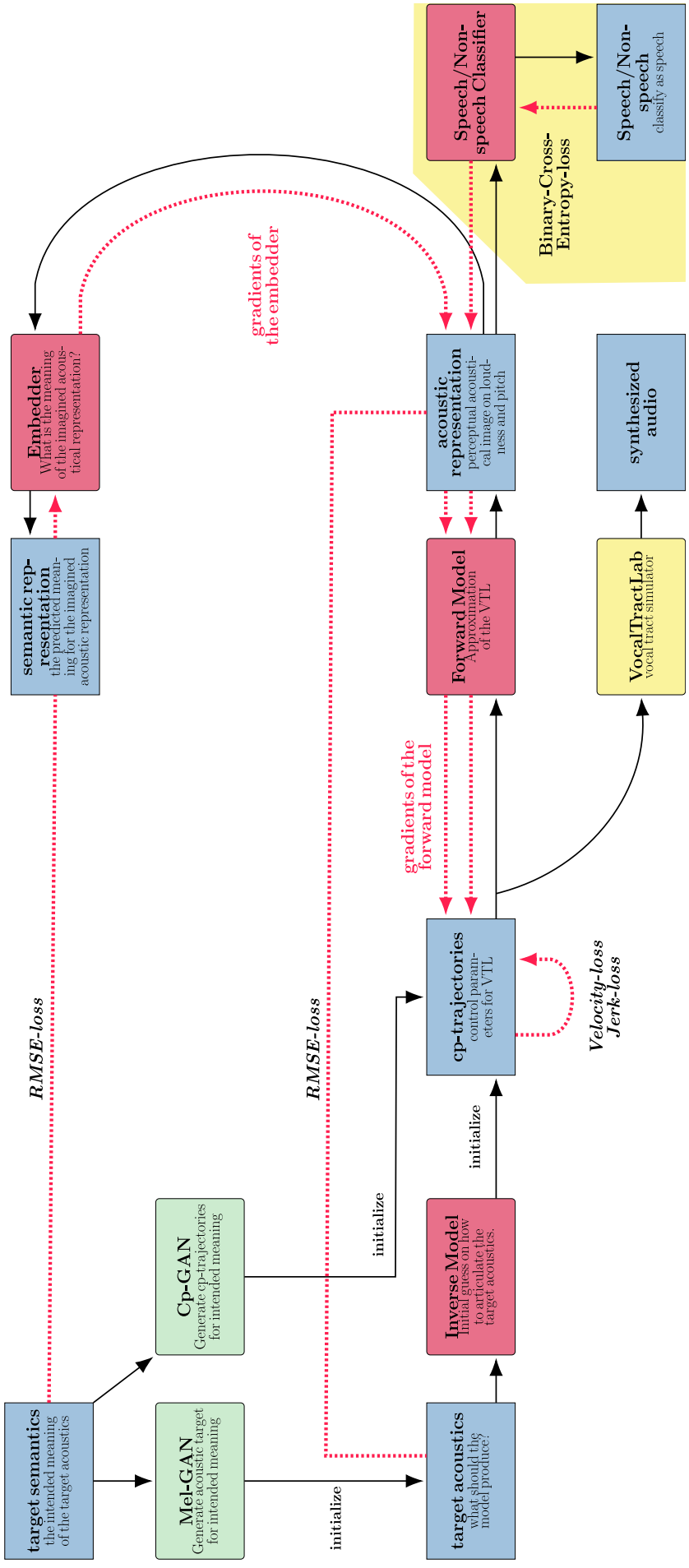

Figure 2 – The speech/non-speech classifier is added to PAULE. The classifier predicts how speech-like the acoustic representation is. As a speech-like synthesis is desired, the error compared to the speech-label is backpropagated along the gradients (red dashed lines). The new data structure and model is highlighted with a yellow background.

3 Training and Results

The model is trained for 100 epochs with a batch size of 64. The AdamW optimizer [11] with an initial learning rate of 0.0001, which gets lowered to 0.00001 after 80 epochs is used. As a loss a binary cross-entropy loss is used with a sigmoid applied before calculating the loss. This leads to logit-like representation at the output layer. During training the training loss decreases from an initial loss of 0.6 to 0.028 for the training loss and to 0.058 for the validation loss. The validation and test accuracy was at 98% for held out data.

These are very good results, for the supervised training of the classifier. These results indicate that either the model is (overly) powerful or that the training data is to easy or both. Evaluating the improvements of the PAULE planning loop is more tricky, as it is computationally expensive and as the planning is not deterministic.

Subjective evaluations of PAULE synthesis with a noisy target acoustic shows only limited improvements using the classifier compared to a PAULE synthesis without the classifier. One reason might be that we furthermore observed that many noisy sound samples that contain only small amounts of speech are classified as speech by the classifier. Instead of using a speech/non-speech classifier a classifier that classifies between own speech and speech by a different speaker might be more suitable. Some ideas on how to improve on these results are discussed in the following.

4 Discussion

Training a speech classifier and using the error in the speech/non-speech goal space as a source to improve the planning in PAULE did only yield limited improvements with the current implementation of the speech/non-speech classifier. This – most likely – can be mainly attributed to the fact that the training data is too easy to classifier. Environmental sounds are too different from speech to encode a strong meaningful gradient into the classifier. Furthermore, manual testing by us indicates that environmental sounds are classified as speech, whenever speech segments are present in the acoustics. If the classifier classifies everything as speech, which has some speech-like signal in it, it cannot give an error anymore to improve on this signal. This can be seen as the classifier is already satisfied with the speech-like-ness in the signal. This is not the behaviour we are seeking for though.

There are two lessons that can be learned here:

First, it is important to select the training data appropriately to the task at hand. In retrospect, it would have been wiser to only use very clean studio recordings and speech synthesis by the VTL with the segment-based approach as the speech class and use besides the environmental sounds, different noise profiles as well as speech with background noise as "non-speech". This should help the classifier to distinguish between speech that contains substantial amounts of noise and hopefully shifts the classification boundary more into a regime where the classifier distinguishes clear speech from non-clear speech instead of speech from non-speech.

Second, the accuracy of the classifier is not the most important metric. It is far more important that the gradients of the classifier after training are informative to the planning process in PAULE. Therefore, the generalizability and robustness of the classifier is more important than its accuracy. It is difficult to quantify this informative gradients though, which makes it easy to first optimize the model architecture of the classifier towards high accuracy. In a next step, we therefore want to explore an LSTM (long-short term memory) architecture [12] and a simple multilayer perceptron (MLP) with a mean loss over the sequence length. Both of these are expected to yield a lower accuracy, but hopefully better gradients for the planning process. The better suited gradients are expected as the models are more constraint in their capabilities.

Under specific circumstances, the MLP can even been seen as a classical logistic regression, which might be the most robust model for the task at hand.

Analogous to the speech/non-speech classifier other goal spaces like a goal space of *loudness*, the *emotional category* of the language or even *speech type* (model, singing, whispering), or imitating *voices* can be implemented. These imitation of voices is different to voice-cloning approaches, as PAULE is restricted to the geometry of the speech organ implemented by the VTL. Inferring a different vocal tract geometry from speech only is a different and challenging task and out of scope for PAULE. Inferring the vocal tract geometry would be necessary to do real articulatory voice cloning. As a guidance of creating other goal spaces: The lower dimensional the space is the better PAULE should be able to adapt to the desired outcomes. Explicitly defining a goal space in terms of a classifier is different to the introduction of the somatosensory pathway, which introduced a new intermediated data representation [13]. In order to achieve good results, it is important to select good training data and a simple classifier that gives informative gradient. This gradient allows PAULE to change the cp-trajectories in the motor space into the direction of the desired outcome in the newly defined goal space.

The planning in PAULE uses a fixed weighting scheme in its planning loss. This should be – theoretically speaking – inferior to an adaptive weighting scheme, which focusses on the goal space where a change in the cp-trajectories has the biggest impact on the loss. At the same time the weighting cannot be unconstrained as the different components of the planning loss are contradicting each other. For example, there is only one solution for a minimal velocity and a minimal jerk, the flat curve of a constant value over time. For all time-varying trajectories either the velocity is non-zero or the jerk is non-zero or both. This leads to the interesting optimization problem: On the one hand, an adaptive weighting scheme is desired for faster planning. On the other hand, the adaptive weighting scheme should still comply to the predefined weightings of the goal space after planning is finished. This is analogous to using a stochastic gradient decent (SGD) algorithm, where an adaptive Adam [14] or AdamW [11] optimizer in most cases has a faster convergence rate.

Some challenges remain in this ongoing research project [15, 16, 17]. In future research, we will test, if a more narrow noisy-vs-clean speech classifier can mitigate the problem of noisy target acoustic outlined here. Furthermore, we work on improvements to the real time factor of PAULE, which is at the moment around 3,000. A real time factor of 3,000 means that for one second of speech 3,000 seconds or 50 minutes of compute time are needed to solve the inverse problem with PAULE. Lastly, with different internal models hopefully the internal self-mumbling within PAULE can be reduced or eliminated. At the moment PAULE needs to mumble substantially to itself to keep the forward model in sync with the VTL speech synthesizer. This self-mumbling is computational expensive and conceptually undesired. A last challenge, which we already started addressing [18], is creating an objective comparison between different articulatory speech synthesis models preferable those, which use the VTL synthesizer as their synthesis model.

4.1 Conclusion

A speech/non-speech classifier helps partially to mitigate the noise in the resynthesized signal, if noise is present in the target acoustic. The theory and rational build up here is nevertheless helpful to pursue this mitigation further. In a next step with a clear-speech/noisy-speech classifier. Furthermore, rational can straightforwardly be adapted to other goal space domains like emotional state of the speech to be produced.

Acknowledgement: This work started as a student project and I thank André Märtins, Dustin Popp and Lukas Bächle for their explorations and work on first implementations. Kon-

stantin Sering is funded by the DfG grant 527671319 („Komplexe Wörter im Kontext“). Konstantin Sering is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

References

[1] SCHMIDT-BARBO, P., S. OTTE, M. V. BUTZ, R. H. BAAYEN, and K. SERING: *Using semantic embeddings for initiating and planning articulatory speech synthesis*. In O. NIEBUHR, M. S. LUNDMARK, and H. WESTON (eds.), *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2022*, pp. 32–42. TUDpress, Dresden, 2022.

[2] SERING, K.: *Predictive Articulatory speech synthesis Utilizing Lexical Embeddings (PAULE)*. Universität Tübingen, Tübingen, 2023. doi:10.15496/publikation-90142.

[3] BIRKHOLZ, P.: *Modeling consonant-vowel coarticulation for articulatory speech synthesis*. *PLOS ONE*, 8(4), pp. 1–17, 2013. doi:10.1371/journal.pone.0060603.

[4] BIRKHOLZ, P.: 2018. URL `http://www.vocaltractlab.de/index.php?page=vocaltractlab-about`.

[5] GAO, Y., S. STONE, and P. BIRKHOLZ: *Articulatory copy synthesis based on a genetic algorithm.* In *INTERSPEECH*, pp. 3770–3774. 2019.

[6] SERING, K., N. STEHWIEN, Y. GAO, M. V. BUTZ, and H. BAAYEN: *Resynthesizing the geco speech corpus with vocaltractlab*. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pp. 95–102, 2019.

[7] GRAVE, E., P. BOJANOWSKI, P. GUPTA, A. JOULIN, and T. MIKOLOV: *Learning word vectors for 157 languages*. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[8] VASWANI, A., N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, and I. POLOSUKHIN: *Attention is all you need*. *Advances in neural information processing systems*, 30, 2017.

[9] ARDILA, R., M. BRANSON, K. DAVIS, M. HENRETTY, M. KOHLER, J. MEYER, R. MORAIS, L. SAUNDERS, F. M. TYERS, and G. WEBER: *Common voice: A massively-multilingual speech corpus*. *arXiv preprint arXiv:1912.06670*, 2019.

[10] PICZAK, K. J.: *ESC: Dataset for Environmental Sound Classification*. 2015. doi:10.7910/DVN/YDEPUT.

[11] LOSHCHILOV, I. and F. HUTTER: *Fixing weight decay regularization in adam*. *CoRR*, abs/1711.05101, 2017. URL `http://arxiv.org/abs/1711.05101`. `1711.05101`.

[12] HOCHREITER, S. and J. SCHMIDHUBER: *Long short-term memory*. *Neural computation*, 9, pp. 1735–80, 1997. doi:10.1162/neco.1997.9.8.1735.

[13] SERING, K. and P. SCHMIDT-BARBO: *Somatosensory feedback in PAULE*. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2023*, pp. 119–126, 2023.

[14] KINGMA, D. P. and J. L. BA: *Adam: A method for stochastic optimization*. *3rd International Conference for Learning Representations*, abs/1412.6980, 2015.

[15] SERING, K., P. SCHMIDT-BARBO, S. OTTE, M. V. BUTZ, and H. BAAYEN: *Recurrent gradient-based motor inference for speech resynthesis with a vocal tract simulator*. In *12th International Seminar on Speech Production*. 2020.

[16] SCHMIDT-BARBO, P., E. SHAFAEI-BAJESTAN, and K. SERING: *Predictive articulatory speech synthesis with semantic discrimination*. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2021*, pp. 177–184, 2021.

[17] SCHMIDT-BARBO, P., S. OTTE, M. V. BUTZ, R. H. BAAYEN, and K. SERING: *Using semantic embeddings for initiating and planning articulatory speech synthesis*. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2022*, pp. 32–42, 2022.

[18] SERING, K. and P. SCHMIDT-BARBO: *Articubench - An articulatory speech synthesis benchmark*. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2022*, pp. 43–50, 2022.