ARTICUBENCH – AN ARTICULATORY SPEECH SYNTHESIS BENCHMARK

Konstantin Sering, Paul Schmidt-Barbo

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

Abstract: The articubench benchmark is a collection of tasks, datasets and scores or metrics in the articulatory speech synthesis domain. The benchmark allows the comparison of different control models for the VocalTractLab speech synthesizer. Articubench is work in progress with the newest results and comparisons of models available at `https://github.com/quantling/articubench`. This publication states the motivation, goals and rationale behind articubench. Tasks, datasets and scores are presented.

1 Motivation

A well-defined task on a freely available common dataset can boost the development of models solving this task and summarising the dataset in a meaningful way. In the domain of image recognition the MNIST dataset [1] constituted an early baseline, which was later massively upscaled by the ImageNet dataset [2]. To the best of our knowledge, the domain of articulatory speech synthesis lacks such an easily accessible dataset and task.

The domain of articulatory speech synthesis is at least threefold with three aspects of evaluation. The first aspect is to perform the articulation in a human-like low effort way. The second aspect is to convey the meaning or semantics of the articulated utterance, which boils down to a classification problem and associated task. The third aspect is the acoustic similarity and quality of the articulated utterance.

These three aspects will be addressed with the python package *articubench*, with which we take a first step to fill this gap of a missing freely available common dataset. Together with specific setup instructions, the performance of a new articulatory speech synthesis control model can be compared to already existing ones. The package combines and publishes data and analysis scripts from earlier publications [3], [4] and adds data from the Mozilla Common Voice project [5], from the KEC [6] and from the GECO corpus (small portion of the transcription only; [7]). The benchmark uses the VocalTractLab (VTL) speech synthesizer as its vocal tract model [8].

The benchmark calculates a total score based on subscores for three different groups of evaluation scores. The idea is to give a reproducible, comparable measure to improve the development of control models by revealing advantages and disadvantages in the proposed tasks.

In the following sections the three different tasks (copy-synthesis, semantic-acoustic, semantic-only) and the three groups of scores (articulatory, acoustics, semantics) with their associated subscores are defined and explained. This is followed by a short presentation of the used datasets. In the end, the limitations of this benchmark and future plans are discussed.

The newest scores and comparisons of control models will be published alongside the package[1] and on the Python package index[2]. Furthermore, the code used in this benchmark can be a

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

[2] `https://pypi.org`

good starting point for scientists, who want to compare electromagnetic articulography (EMA) or mid sagittal tongue contour ultra sound recordings with the virtual tongue movements of the VTL synthesis.

1.1 VocalTractLab (VTL)

The articubench benchmark uses the VocalTractLab (VTL) speech synthesiser api with the JD3 speaker in version 2.5.1quantling. In this configuration, the VTL synthesizer takes a sequence of 30 control parameters (19 control parameters for the vocal tract shape and 11 control parameters for the glottis configuration) defined every 110 audio samples (2.5 ms) as input and emits a 44,100 Hz mono audio signal. The quality of the provided control parameter sequence (cp-trajectory) determines the quality of the resulting audio signal and is, therefore, the basic building block of the articubench score. In order to compete in the benchmark each control model is expected to output a valid cp-trajectory. Note that the VTL comes with a rich frontend to orchestrate gestural scores and visualize results. This frontend is not used here.

2 Tasks

In the field of articulatory speech synthesis and articulatory phonology, one of the main tasks is to record, systematize and understand the human speaking process. Particular focus is placed on the use of the articulators, which show a wide variability across different speakers but also within a single speaker. Still human articulation produces intelligible speech with seemingly low effort. As a first step to capture this huge challenge, which needs to be clarified and specified furthermore, we define and implement three word-level tasks.

All tasks are linked to but should not be confused with the evaluation scores and metrics used in the benchmark. The task itself defines the given inputs and the expected outputs, which is different to the quality measures described in the scores section.

2.1 Acoustic-only (Copy-Synthesis) Task

In the acoustic-only or copy-synthesis task only a target (human) audio recording with no outside information on the intended meaning is given. The control model is supposed to return cp-trajectories of the same duration as the target recording. Meaningful and important scores are those from the articulatory and acoustic domain, whereas the scores from the semantic domain come second.

2.2 Semantic-only Task

The semantic-only task starts with a target semantic embedding vector and seeks to find cp-trajectories that produce a synthesis, which the embedder embeds closely to the target semantic embedding vector. For this task the acoustic metrics receive less attention as there is no objective acoustic attached to each semantic embedding (a single to many problem). The metrics of the articulator movements and the semantics, however, are still valid.

2.3 Semantic-Acoustic Task

In the semantic-acoustic task, both a (human) recording of a single word and a target semantic embedding vector are given. The goal of the control model is to find a set of cp-trajectories that jointly match the recording and the semantic embedding vector as close as possible. All metrics and associated subscores are equally relevant to this task.

3 Scores

Where the tasks define what the control model is required to do, the scores define different metrics on how the performance of the control model is evaluated. Thereby the scores and subscores are defined in a way that a higher score means a better control model. Each subscore usually lies between 0 and 100.

In the domain of articulatory speech synthesis we distinguish three important groups of scores. The first group of subscores measures the quality of the *articulatory movements*. As there is high variability in human articulatory movements there is no single gold standard movement of the articulators given a specific target acoustics. Nevertheless, the articulators of the VTL should follow some distributional properties of the human articulators and should be comparable to sensor measurements of the human tongue. The articubench benchmarks evaluates the velocity and jerk distribution of the cp-trajectories. Furthermore, it compares the virtual tongue movement to electromagnetic articulography (EMA) data at the tongue tip and the tongue body and compares the virtual tongue height to the human tongue height using ultra sound measurement of the mid sagittal plane. The second group of subscores belongs to the *semantic domain*. This is implemented by looking at the closeness to a target semantic vector embedding and the classification rank in a single word classification setup. The third group of subscores belongs to the *acoustic domain*. With these subscores the similarity between synthesis and target audio recording are evaluated by comparing a loudness envelope and two log-mel spectrograms. Subscores for the f0 and formant transitions should be added later on. Each group score as well as the total score are calculated as the sum of all subscores.

To calculate the scores of a control model, the predicted cp-trajectories serve as input to the VTL. After deriving the corresponding audio and virtual tongue movements, scores are calculated in the following way. First an error on each token is calculated and averaged over all tokens in the dataset. In a next step the average error is normalised by the average error of the baseline model in corresponding subscore. Afterwards, the resulting normalised average error is subtracted from 1 and multiplied by 100. This ensures that having no error results in a subscore of 100 and an error of the size of the baseline model results in a subscore of 0. In the equations this normalization is denoted by the text *baseline model* in the denominator. Most errors are calculated by computing the root mean squared error (RMSE) between resulting synthesis and the target.

The the total score S_{total} and the three group scores $S_{\text{articulator}}$, S_{acoustic}, S_{semantic} are defined the following way:

$$S_{\text{total}} = S_{\text{articulatory}} + S_{\text{semantic}} + S_{\text{acoustic}} \qquad S_{\text{semantic}} = S_{\text{sem\_dist}} + S_{\text{sem\_rank}}$$

$$S_{\text{articulatory}} = S_{\text{tongue\_height}} + S_{\text{ema}} + S_{\text{vel\_jerk}} \qquad S_{\text{acoustic}} = S_{\text{loudness}} + S_{\text{spectrogram}}$$

3.1 Articulatory Scores

3.1.1 Tongue height

The highest point of the tongue can by easily determined by a mid sagittal tongue contours measurement using ultra sound. As the highest point of the tongue is defined for every time point, it is possible to apply a point-wise measure before averaging over all time points and tokens. In order to extract tongue information from VTL, we export the mid sagittal plane as an svg image and rotate and shift the svg image to match a standard orientation of an ultra sound image. We extract the highest point of the tongue line in the svg. Scores are normalised to the baseline model.

$$S_{\text{tongue\_height}} = 100 \cdot \left(1 - \frac{mean_{\text{token}}(RMSE(\text{height}_{\text{synthesis}}, \text{height}_{\text{ultrasound}}))}{\text{baseline model}}\right)$$

3.1.2 Virtual EMA Points

Electromagnetic articulography (EMA) allows to track sensors glued on the tongue or other articulators of the human speech organ during speech production. Sensor can be tracked over time together with their prepared and "registered" location. This allows for a direct comparison to virtual EMA sensors of the VTL. In articubench we focus on the tongue tip and tongue middle sensor with an RMSE along the bottom-top z and back-forth x movement. Scores are normalised to the baseline model.

$$S_{\text{ema}} = 100 \cdot \left(4 - \frac{mean_{\text{token}}(RMSE_{\text{TT,x}} + RMSE_{\text{TT,z}} + RMSE_{\text{TM,x}} + RMSE_{\text{TM,z}})}{\text{baseline model}}\right)$$

$$RMSE_{\text{TT,x}} = RMSE(\text{tongue\_tip}_{\text{synthesis,x}}, \text{tongue\_tip}_{\text{ema,x}})$$

The remaining values $RMSE_{\text{TT,y}}$, $RMSE_{\text{TM,x}}$, and $RMSE_{\text{TM,y}}$ are calculated respectively.

3.1.3 Velocity and Jerk of Control Parameters

Due to the different size of the vocal cavity and the difficulty to define a standardized origin and orientation for the articulator movements, we focus on some distributional movement patterns of the vocal tract and glottis cp-trajectories in terms of confidence for human-like statistical properties. This means especially that the cp-trajectories are smooth, i. e. do not have any jumps nor sharp corners, and don't have extreme velocities (changes in position) nor extreme jerks (changes in acceleration or applied force). In contrast to the other scores, velocities and jerks are normalised by the cp-trajectories of the resynthesised GECO corpus as the baseline model has a jerk and velocity of zero.

$$S_{\text{vel\_jerk}} = 100 \cdot \left(2 - \frac{mean_{\text{token}}(max(\text{velocity}_{\text{synthesis}}))}{max(\text{velocity}_{\text{GECO}})} - \frac{mean_{\text{token}}(max(\text{jerk}_{\text{synthesis}}))}{max(\text{jerk}_{\text{GECO}})}\right)$$

3.2 Semantic Scores

3.2.1 Semantic Embedding

The first score for the semantic is the distance between the semantic embedding vector of the synthesis compared to the target semantic embedding vector normalised to the distance between the baseline model to the target semantic embedding vector.

$$S_{\text{sem\_dist}} = 100 \cdot \left(1 - \frac{mean_{\text{token}}(RMSE(\text{semantic\_vector}_{\text{synthesis}}, \text{semantic\_vector}_{\text{target}}))}{\text{baseline model}}\right)$$

3.2.2 Classification Rank

The rank of a target semantic vector is determined by calculating the correlation of the embedded vector to 4311 reference semantic vectors including the target and ranking them from highest to lowest. The rank of the target semantic vector is used to calculate a score:

$$S_{\text{sem\_rank}} = 100 \cdot \left(1 - \frac{mean_{\text{token}}(rank_{\text{target}} - 1))}{4311}\right)$$

3.3 Acoustic Scores

3.3.1 Loudness over Time

One of the broadest measures to check, if the acoustics of the synthesis roughly matches the target recording, is to compare the loudness envelope. To do so, we calculate the loudness every 220 samples over a 1024 sample window by summing all log-mel spectrogram entries (see below) for each time slice:

$$S_{\text{loudness}} = 100 \cdot \left(1 - \frac{mean_{\text{token}}(RMSE(\text{loudness}_{\text{synthesis}}, \text{loudness}_{\text{recording}}))}{\text{baseline model}}\right)$$

3.3.2 Log-Mel Spectrogram

The log-mel spectrogram roughly approximates the signal decomposition performed by the human ear as it maps the frequency to pitch and the energy or magnitude to loudness. We use a Mel spectrogram with 60 banks in the frequency range from 10 to 12000 Hz, a time shift of 110 samples (2.5 ms) and an aggregation window for the Fourier transform of 1024 samples (23.2 ms). From the magnitude values of the Mel spectrogram the logarithm is computed to map it from a physical energy measure to a perceptual loudness measure on a db-scale. The resulting loudness values are finally mapped to the $[0, \inf)$ interval where 1 is a loud tone and 0 corresponds to silence.

$$S_{\text{spectrogram}} = 100 \cdot \left(1 - \frac{mean_{\text{token}}(RMSE(\text{spectrogram}_{\text{synthesis}}, \text{spectrogram}_{\text{recording}}))}{\text{baseline model}}\right)$$

4 Datasets

The benchmark comes in three sizes: a *tiny* one with one to two tokens per dataset; a *small* one with around 30 tokens per dataset and a *normal* one with around 1000 tokens per dataset or all available tokens in the dataset. With the three different sizes we account for the need of testing if a control model performs at all (tiny), brings comparable results (small) and is better or worse along the language statistics in a statistically robust way (normal).

The articubench focusses on the German language and therefore only German datasets are used so far.

4.1 KEC

From the KEC [6] we select 1779 /ja/ and /halt/ word tokens spliced out of a conversation between two of 79 acquainted, native speakers of Southern German. Each conversation lasted for one hour and was hold in separate booths. Beside audio, EMA data was recorded from which we use the tongue tip and tongue body sensors. Manual annotation at the word level is provided while automatic annotation at the segment level as well as an automatic morphological tagging is added.

4.2 Mozilla Common Voice

Mozilla Common Voice [5] recordings are mostly read out speech from a crowd source project, which are freely available. For articubench we use a small portion of the German part of the Common Voice corpus, which we aligned with the Montreal Forced Aligner (MFA) [9] to cut out single word tokens. The dataset is less natural compared to the KEC and GECO but more natural than professional speech from professional speakers in radio broadcasts or TV shows.

4.3 GECO

The GECO [10] contains 46 dialogues of approx. 25 minutes length between previously unacquainted female subjects. Similar to the KEC, conversational speech comes prealigned on the word and phone segment level. Articubench uses the phone segment transcriptions files of 1000 words.

4.4 babibabubaba

In the babibabubaba dataset [4] ultra sound recordings of the mid sagittal plane of the tongue were measured for the artificial non-words /babi/, /babu/, /baba/ to create human data on coarticulation effects of formant transition in the /a/ sound.

5 Control Models

A control model competing in articubench benchmark needs to implement an interface receiving either an audio recording, a semantic embedding vector and a duration, or all three. The output must consists of a cp-trajectory of the requested duration. Furthermore, if implemented as a machine learning model, it needs to state the number of trainable parameter and its energy consumption for training. Table 1 shows an overview of different models present in articubench.

5.1 Baseline Model (Schwa-Model)

The baseline model always returns the neutral gesture of the JD3 speaker of the VTL. With the neutral gesture the VTL produces a constant Schwa-sound.

5.2 Segment-based Model

The segment based synthesis uses phone segments and their corresponding duration to blend gestural scores of the JD3 speaker together overtime. The blended gestures result in smooth cp-trajectories. For the acoustic-only task the given acoustics has to be labeled with a sampa-phone transcription. After aligning the phone segment using the MFA, [9], cp-trajectories can be generated.For the semantic-only task, sampa transcription for the target word need to be looked up in a data base and standard phone durations are scaled to the desired total length of the word. In the semantic-acoustic task, the target semantic embedding vector is used for the sampa lookup, while durations of the phone segments are derived from an alignment [9].Although the segment based model has no trainable parameters and is fast to execute, it relies on a substantial amount of handcrafting, knowledge and fine tuning.

5.3 Inverse Model / Cp-GAN

The inverse model is a direct mapping of a log-mel spectrogram to cp-trajectories. In the PAULE model [11] it is used as one option to initialize the cp-trajectories. However, it can be used stand-alone for the copy-synthesis task as well. As a second option of initialization, the PAUL model introduces a Cp-GAN, generating cp-trajectories from a semantic vector. Again the initialisation can be evaluated as a stand-alone model.

5.4 Predictive Articulatory speech synthesis Utilising Lexical Embeddings (PAULE)

The complete PAULE control model[3] [11] is the heaviest of the control models showcased here. It uses the inverse model or the Cp-GAN as initialisation before it iteratively plans even better cp-trajectories, which is computational intense.

Table 1 – Comparing the PAULE, inverse, segment-based and baseline control models along different model properties. The memory demand includes the python binaries (200 MB). The segment model needs an embedder and the MFA [9] in its pipeline, for which the data is given in parenthesis.

Property	PAULE	Inverse	Seg-Model*	Baseline-Model
Trainable parameters [million]	15.6	2.6	0 (6.6 + MFA)	0
Execution time [seconds]	200	0.2	0.5 (0.3 + MFA)	< 0.0001
Memory demand [MB]	5600	5000	2 (5100 + MFA)	200
Energy used for Training [kWh]	393	1.9	0.0 (7.6 + MFA)	0.0

6 Limitations, Next Steps, & Conclusion

The articubench benchmark is limited to the German language, neither English nor any tonal language are used so far. It focuses on evaluating single spliced-out words from read-out or conversational speech instead of whole phrases. Comparing scores on the benchmark for different models might overlook important theoretical and practical differences between the models. Intelligibility of the speech cannot be easily compared, as there is no easy automated way to do this so far. The benchmark includes a pretrained embedder model that maps log-mel spectrograms to semantic embeddings. Although needed to be included with fixed weights to make the benchmark deterministic and replicable, this embedder model is only one possible mapping between an audio signal and semantic embedding vectors and therefore has no ground truth justification. Furthermore, some of the scores that approximating human judgements are a lot less precise and at the same time more sensitive to noise compared to the human perception of speech. However, as a benchmark is always somehow an arbitrary choice of comparison, articubench can give a quality overview of control models with a coarse grained insight on strengths and weaknesses. In order to evaluate and understand small differences in control model quality a human judgement is still necessary.

Still missing are two important acoustic scores: first, the metric of formant transitions in the /babi/, /babu/, /baba/ utterances and second, matching changes in the fundamental frequency (f0), which might be an especially interesting score for tonal languages. Both of these require a robust and automatic way of extracting these measures in python.

Furthermore, as most human conversations use more than single words and articulatory patterns can stretch longer periods than single words, it is favorable to complement the articubench benchmark with whole phrase dataset and tasks. Plans include to add an English and Mandarin speaking dataset. On the control model side, it would be nice to have a DIVA [12] like control model for VTL as well as the generative optimisation approach developed by Gao et al. [13].

In conclusion, we propose an articulatory benchmark, which compares control models for the VocalTractLab speech synthesiser along the articulatory, semantically, and acoustically domain. Suggestions on how to improve articubench are highly appreciated and will be incorporated in future version of articubench published as free and open source software at `https://github.com/quantling/articubench`.

[3] https://github.com/quantling/paule

Acknowledgments This research was supported by an ERC Advanced Grant (no. 742545) awarded to R. Harald Baayen. Konstantin Sering and Paul Schmidt-Barbo are members of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

References

[1] LECUN, Y.: *The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/*, 1998.

[2] DENG, J., W. DONG, R. SOCHER, L.-J. LI, K. LI, and L. FEI-FEI: *Imagenet: A large-scale hierarchical image database*. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[3] SERING, K. and F. TOMASCHEK: *Comparing kec recordings with resynthesized ema data*. In A. WENDEMUTH, R. BÖCK, and I. SIEGERT (eds.), *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*, pp. 77–84. TUDpress, Dresden, 2020.

[4] SERING, K., F. TOMASCHEK, and M. SAITO: *Anticipatory coarticulation in predictive articulatory speech modeling*. In S. HILLMANN, B. WEISS, T. MICHAEL, and S. MÖLLER (eds.), *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2021*, pp. 208–215. TUDpress, Dresden, 2021.

[5] 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.

[6] ARNOLD, D. and F. TOMASCHEK: *The karl eberhards corpus of spontaneously spoken southern german in dialogues - audio and articulatory recordings*. In C. DRAXLER and F. KLEBER (eds.), *Tagungsband der 12. Tagung Phonetik und Phonologie im deutschsprachigen Raum.*, pp. 9–11. Ludwig-Maximilians-Universität München, 2016.

[7] 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.

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

[9] MCAULIFFE, M., M. SOCOLOF, S. MIHUC, M. WAGNER, and M. SONDEREGGER: *Montreal forced aligner: Trainable text-speech alignment using kaldi*. In *Interspeech*, vol. 2017, pp. 498–502. 2017.

[10] SCHWEITZER, A. and N. LEWANDOWSKI: *Convergence of articulation rate in spontaneous speech*. In *INTERSPEECH*, pp. 525–529. 2013.

[11] SCHMIDT-BARBO, P., S. OTTE, M. V. BUTZ, H. BAAYEN, and K. SERING: *Using semantic embeddings for initiating and planning articulatory speech synthesis*. In *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2022*. TUDpress, Dresden, 2022.

[12] TOURVILLE, J. A. and F. H. GUENTHER: *The diva model: A neural theory of speech acquisition and production*. *Language and cognitive processes*, 26(7), pp. 952–981, 2011.

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