CASTING TO CORPUS: SEGMENTING AND SELECTING SPONTANEOUS DIALOGUE FOR TTS WITH A CNN-LSTM SPEAKER-DEPENDENT BREATH DETECTOR

Éva Székely, Gustav Eje Henter, Joakim Gustafson

Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden

ABSTRACT

This paper considers utilising breaths to create improved spontaneous-speech corpora for conversational text-to-speech from found audio recordings such as dialogue podcasts. Breaths are of interest since they relate to prosody and speech planning and are independent of language and transcription. Specifically, we propose a semi-supervised approach where a fraction of coarsely annotated data is used to train a convolutional and recurrent speaker-specific breath detector operating on spectrograms and zero-crossing rate. The classifier output is used to find target-speaker breath groups (audio segments delineated by breaths) and subsequently select those that constitute clean utterances appropriate for a synthesis corpus. An application to 11 hours of raw podcast audio extracts 1969 utterances (106 minutes), 87% of which are clean and correctly segmented. This outperforms a baseline that performs integrated VAD and speaker attribution without accounting for breaths.

Index Terms— Spontaneous speech, found data, speech synthesis corpora, breath detection, computational paralinguistics

1. INTRODUCTION

Speech synthesis from recordings of spontaneous conversations has evident potential to be more suitable for interactive settings. However, it is difficult to obtain suitable data for training such systems. Most approaches to conversational speech synthesis have relied on small, hand-annotated corpora [1, 2, 3, 4], producing no more than 2100 utterances or 75 minutes of speech. Such limited datasets cannot capture the rich prosodic variability of spontaneous speech.

Recent improvements in automatic speech recognition (ASR) accuracy (e.g., [5]) open up the possibility to leverage large amounts of found, unlabelled, spontaneous speech audio for speech synthesis. However, to achieve acceptable synthesis quality it is crucial that only clean and accurately-transcribed utterances are used in training [6], a requirement which is even more important for speech synthesis from found data [7, 8, 9]. Leading ASR services do not consistently transcribe phenomena like hesitations, backchannels, silences, and filled pauses, and their segmentation, diarisation, and overlapped speech detection were not designed with conversational TTS in mind. There is therefore still a need for careful custom processing of found audio recordings to enable successful synthesis.

In this work, we propose to automatically segment and separate speakers in found-speech recordings for speech synthesis corpus creation through the use of speaker-specific breath event detection. Specifically, we train a neural breath event detector based on annotating breath events and silences a small part of the data, and use the trained detector to select individual segments delineated by breath events, also called *breath groups*, as training utterances for the synthesiser. The main contributions are thus:

1. Training speaker-specific breath-event detectors inspired by recent advances with deep image analysers in computational paralinguistics [10].
2. Using automatically-detected breath groups to define speaker-specific input utterances for speech synthesis, enabling segmentation and speaker allocation without a transcript.

Sec. 2 outlines the significance of breath events and image-based methods in speech and paralinguistics, while Sec. 3 describes our corpus-creation methodology in detail. An application in Sec. 4 to 10 episodes of a 150+ episode two-person audio-only public-domain podcast illustrates the approach and extracts more single-speaker conversational speech than used in prior work [1, 2, 3, 4].

2. BACKGROUND

2.1. Breath Events in Speech Analysis and Technology

The utility of breath events has been a focus of recent research attention in areas as diverse as speech synthesis [11], ASR [12, 13], speech diagnostics [14], and speech analysis [15]. In particular, Fukuda et al. [13] noted a 3.8% reduction in character error rate when segmenting speech based on detected breath events prior to ASR. A segmentation that improves ASR accuracy is obviously desirable for text-to-speech (TTS) applications involving automatic transcriptions of found speech audio. There are, however, additional reasons to believe that TTS can benefit from breath-based segmentation, particularly in the case of spontaneous speech recordings.

Spontaneous speech has no punctuation and adheres less to the rigid grammar of (read-aloud) written language. While most synthesis frameworks assume that corpora contain isolated, single-sentence utterances – e.g., for part-of-speech tag features [16] – it is not always easy to partition conversational speech into valid and well-defined sentences in the linguistic sense. However, there is a close correspondence in standard TTS corpora between linguistically-defined sentences and speaker breaths, as most utterances in these corpora consist of a single breath group. It has also been found [17] that speakers' respiratory patterns are involved in the speech planning process in spontaneous conversations. Breaths are furthermore highly correlated with major prosodic breaks [18] and turn-taking behaviour [19]. Breath groups therefore seem like a compelling way to segment continuous speech recordings into prosodically-consistent speaker-specific utterances on which conventional TTS can be trained. Breaths are also attractive for this segmentation since they are language- and transcription-independent. Additionally, modelling inhalation pauses was found to improve TTS quality ratings is [11]. We hypothesise that, like ASR [13], conversational TTS will benefit from breath-group segmentation.

This research was supported by the Swedish Research Council Project Incremental Text-To-Speech Conversion VR (2013-4935) and by the Swedish Foundation for Strategic Research project EACare (RIT15-0107).

Author postprint of paper presented at ICASSP 2019, Brighton, UK. Digital Object Identifier 10.1109/ICASSP.2019.8683846

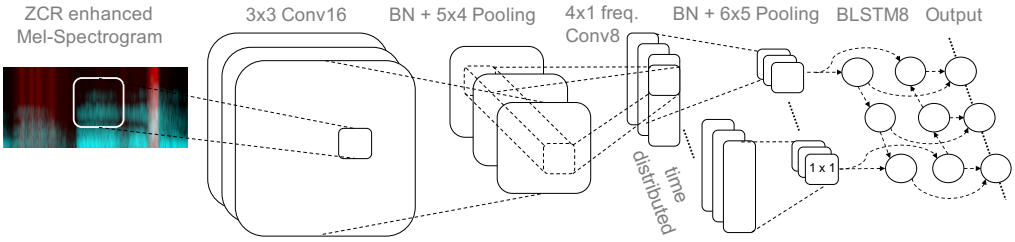

Fig. 1. Schematic illustration of the architecture of the classifier, along with selected layer sizes.

2.2. Computational Paralinguistic Event Detection

Breathing is not part of the linguistic message of speech, so automatic breath-event detection falls under the domain of computational paralinguistics. While contemporary breath-detection (e.g., [13]) commonly uses shallow classifiers, image-processing-based methods using deep learning are now the state-of-the-art on many paralinguistic tasks [10]. Examples include classifying rare acoustic events [20], speech-based emotion recognition [21], and snore classification [22], snores being a type of breath. These use convolutional neural networks (CNNs) applied to log-magnitude spectrograms, mimicking iconic deep image-processing architectures like ImageNet [23]. Adding recurrent (RNN) connections like in [20] can give longer context-sensitivity and more temporally-consistent frame-level classification for events with widely varying durations, like breaths.

Image-processing methods have been applied to speech segmentation before, specifically language-independent phone segmentation [24], but that work segmented speech according to linguistic content and did not incorporate deep learning. Thus, to the best of our knowledge, deep and image-based methods have not yet been explored for the tasks of breath detection, breath-based speaker allocation, or segmenting speech into utterances. The present work attempts to fill these gaps and employ leading CNN and RNN methods to detect and use breaths in automatic speech processing.

3. METHOD

We now describe our proposed breath-based method for extracting single-speaker utterances from dialogue audio. The approach uses a small amount of coarsely annotated data to train an event detector whose output is used both to segment and select sections of audio for TTS corpus creation. We compare our proposed method against a baseline approach that directly selects audio segments attributed to the target speaker, without regard for breaths.

3.1. Data and Seed Annotation

While the approach outlined in this paper applies to any spontaneous dyadic conversation with good recording quality, we will illustrate our proposal with a concrete application to audio from an untranscribed weekly technology podcast, namely the "ThinkComputers" podcast made available in the public domain via the Internet Archive (archive.org). The recordings contain product reviews and discussions of technology news from two male speakers of American English (here called A and B) mixed into a single audio channel. At the time of writing, over 150 episodes are available online, each about an hour long. As a demonstration, we used the Ogg Vorbis audio (71 kbpbs at 48 kHz) from episodes 140 through 150 in our application.

To train speaker-specific breath event detectors it is necessary to annotate a subset of the audio. Podcast episode 148 (62 minutes) was therefore manually annotated in Praat to indicate audio intervals comprising either inhalation from speaker A, inhalation from speaker B, or silence. Each unbroken segment of audio not belonging to either of these categories was further tagged as being either speech fully belonging to speaker A, speech fully belonging to speaker B, as containing speech from both speakers, or as containing other audio impurities (e.g., laughter). Each time instance in the annotated recording was thus assigned to one of 7 different classes.

Since the annotation did not involve transcription or marking phonetic boundaries, nor precisely delineating overlapping speech, annotation was relatively easy, needing only about 3.5 hours. Statistics for this seed data can be found in Table 1 in Sec. 4. 12 minutes of annotated audio in the middle of the episode were held aside as a validation set, with the unannotated episodes serving as the test set. As speaker A delivered the majority of the product reviews in the annotated episode, with speaker B often acting as a listener and discussion partner, it was decided to use A as the *target speaker* for data extraction, to obtain as much same-speaker data as possible.

3.2. Classifier Architecture and Input Features

The second step of our proposed approach is to build and train a system capable of automatically classifying unannotated frames of audio into the annotated categories. Owing to the success of convolutional neural network architectures based on deep image processing in computational paralinguistics (see Sec. 2.2), we propose to employ similar methods also for our breath-event detection and classification. For our experiment we used a network architecture based on [20], with two convolutional (CNN) layers with batch-normalisation and max-pooling are followed by a bidirectional recurrent (LSTM) layer, and ending in a softmax output layer for the seven different classes. This architecture is illustrated graphically in Figure 1. The recurrent layers allow longer temporal context to be taken into account and were found to substantially increase frame-level classification accuracy in preliminary experiments.

As network input we used log-magnitude spectrogram images extracted from the raw audio. Mel-scaled spectrograms were used since they consistently outperformed the linear frequency scale in preliminary experiments. The spectrogram images were encoded either monochromatically or in RGB using the "viridis" colorway, as the latter was reported to perform better in [22]. However, we also investigated augmenting the frequency-domain spectra with time-domain information to improve classification. In particular, the zero-crossing rate (ZCR) has been shown to be an effective feature for differentiating breath events from unvoiced fricatives [25, 13] and have also been of interest for detecting overlapped speech [26]. It is defined as the number of times the audio waveform changes sign divided by the total number of samples in the window. We added ZCR as another image channel to each spectrogram cell, as illustrated in Figure 2. As seen, ZCR information makes breaths (mid ZCR) and fricatives (high ZCR) more visually distinguishable. See Secs. 4.1 and 4.2 for implementation and results on our sample application.

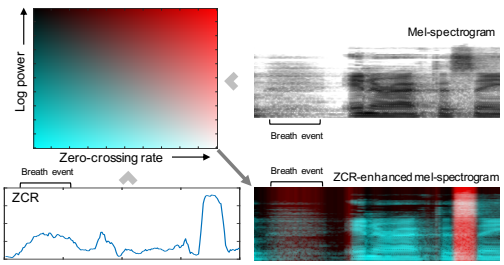

Fig. 2. Example of using ZCR information to enhance a grayscale mel-scale spectrogram containing a breath event.

3.3. Baseline Approach

Since the classifier produces labels that distinguish clean speech from each speaker from other types of annotated audio, a straightforward, naïve approach to TTS corpus creation is to select as a training utterance any unbroken stretch of speech frames (ignoring short silences up to 0.35 s) attributed to the target speaker in the automatic annotation. This ignores breaths, which are merged into the silences. Only segments immediately following silences exceeding 0.35 ms were considered, to avoid producing fragmentary utterances. Conceptually, this is similar to applying voice-activity detection (VAD) to ignore silences and breaths, followed by speaker diarisation to only keep speech segments from the target speaker.

Unfortunately, this simple baseline selector is not likely to give clean and TTS-appropriate segments. Segments are for example less likely to be linguistically or prosodically consistent units, since breaths are ignored (see Sec. 2.1). Another stumbling block for this baseline that the data has a class imbalance problem: speech is much more common than breaths. Any standard classifier is then likely to default to classifying ambiguous frames as speech by the most prominent speaker, owing to the large a-priori probability of this class. This means that there are likely be many instances where audio that is inappropriate for training a single-speaker speech synthesiser is flagged as clean audio from the target speaker. Table 3 confirms this phenomenon on our podcast example data. This bias is undesirable since TTS quality is highly sensitive to inappropriate training data [6, 8, 9]; when starting from a large source of found audio data, we can afford to be selective with what audio is retained, and still arrive at a speech corpus significantly larger than those used for previous work on conversational speech synthesis.

3.4. Proposed Breath-Group Segmentation and Selection

As an alternative to the baseline approach above, we propose to use the classifier output to i) identify single-speaker breath groups as candidate segments for TTS, followed by ii) a sub-selection among these candidate segments based on the probability of each segment being clean speech from the target speaker alone.

As the first sub-step, relevant breath groups are identified in the automatically annotated recordings. We defined these as stretches of target-speaker frames after a target-speaker breath, also including silences up to 0.5 s. (Mixed or non-target-speaker breath groups are not retained, but their position is identified, which can be useful for, e.g., modelling turn-taking in the future.) Both the baseline and the proposed method only considered segments from 1 to 8 seconds long, trimming overlong segments at the last silence before 8 s.

As the second sub-step, extracted candidate segments (target-speaker breath groups) are assessed to rule out interference from the non-target speakers or other undesirable acoustic events, again based on the (largely automated) annotations. This quality control can be performed in several ways. One method is to put a lower bound on p_{worst}, the maximum permissible probability that any given frame in the segment is problematic. Mathematically, we let p_t be the probability that frame t in the audio is acceptable, defined as the sum of probabilities that it is either silence or originates from the target speaker (breath or speech); the probability that a frame is problematic becomes $1 - p_t$. Then, p_{worst} can be computed as

$$p_{\text{worst}}(t_{\text{begin}}, t_{\text{end}}) = \min_{t \in \{t_{\text{begin}}, \ldots, t_{\text{end}}\}} p_t. \quad (1)$$

This worst-frame criterion seems appropriate since problematic events such as overlapping speech resulting from backchannelling (such as *yeah*, *ok*, *uh-um*) are likely to be brief in duration but still should cause the entire candidate segment to be discarded.

Another option is to assume that frames are statistically independent and then evaluate segments based on p_{all}, the probability that they contain exactly zero problematic frames, computed through

$$p_{\text{all}}(t_{\text{begin}}, t_{\text{end}}) = \exp\left(\sum_{t=t_{\text{begin}}}^{t_{\text{end}}} \ln p_t\right). \quad (2)$$

Regardless of which formula is used, the threshold probability for the keep/discard decision can be adjusted to strike a suitable balance between data quantity and quality, e.g., as outlined in Sec. 4.2.

Class	Audio (s) Trn.	Audio (s) Val.	Segments Trn.	Segments Val.	Avg. dur. (s) Trn.	Avg. dur. (s) Val.
1 – Silence	292	73	704	166	0.42	0.44
2 – Breath A	161	33	430	82	0.37	0.40
3 – Breath B	21	9	52	22	0.40	0.40
4 – Speech A	1752	381	746	168	2.35	2.37
5 – Speech B	331	95	187	60	1.77	1.58
6 – Mixed	747	131	247	50	3.02	2.62
7 – Other	45	20	48	20	0.93	0.99
All	3348	741	2414	568	1.39	1.30

Table 1. Statistics for the manually annotated training (Trn.) and validation sets (Val.). "Avg. dur." is the average segment duration.

4. EXPERIMENTS AND RESULTS

This section describes the seed data (Table 1), implementation, and results of our example application to the "ThinkComputers" podcast.

4.1. Implementation and Training Details

Mel-spectrograms were extracted using the librosa Python package with a window width of 20 ms and 2.5 ms hop length. The resulting spectrograms for two seconds of audio have 128×800 pixels. Zero-crossing rates were calculated on the same windows.

The neural network was implemented in Keras following the architecture in Figure 1. The first convolutional layer used 16 2D filters (size 3×3, stride 1×1) and ReLU nonlinearities, followed by batch normalisation and 5×4 max pooling in both time and frequency. The second 2D convolutional layer used 8 filters in the frequency domain (4×1) and ReLU, followed by batch norm and 6×5 max pooling. Due to downsampling by the pooling layers, this produced 40 1×1 cells with 8 channels at a rate of 20 times per second. These were fed into a bidirectional LSTM layer of 8 hidden units in each direction, followed by a softmax output layer.

The network was randomly initialised and trained for 40 epochs to minimise cross-entropy using Adadelta (with default parameters)

Input feature set	All classes Accuracy	Breaths speaker A Precision	Recall
Monochrome	67.5%	90.5%	81.7%
Viridis	69.9%	82.8%	93.9%
Monochrome + ZCR	77.6%	96.3%	95.1%

Table 2. Frame-level classifier performance on the validation set for different spectrogram-based input features.

Class	1	2	3	4	5	6	Sum
1 – Silence	149	1	1	12	-	3	166
2 – Breath A	2	78	-	2	-	-	82
3 – Breath B	4	2	13	-	-	3	22
4 – Speech A	-	-	-	165	-	3	168
5 – Speech B	2	-	-	4	38	16	60
6 – Mixed A+B	-	-	-	23	3	24	50
7 – Other	8	2	2	5	2	1	20
Sum	165	83	16	211	43	50	568

Table 3. Segment-level confusion matrix on the validation set. Each row is the annotated class and each column is the the classifier's prediction. Column "7" is omitted as all its cell counts were zero.

on batches of 16 two-second spectrogram excerpts. The softmax outputs can be interpreted as estimated per-frame class probabilities and used to automatically annotate the held-out episodes.

Prior to further processing by either method, the temporal coherence of the automatic annotations was improved by merging mixed speech after a single-speaker segment into that speaker's speech.

4.2. Analysis of Classifier and Selection Criteria Performance

Table 2 reports the frame-level validation-set performance of classifiers trained on the different input features considered in Sec. 3.2. Evidently, monochrome spectrograms with ZCR outperformed both other variants in all measures considered. The high precision attained is crucial for consistently finding only correct target-speaker breath groups. Hence this feature representation was used in all remaining experiments. Table 3 displays a segment-level confusion matrix for this chosen classifier on the validation set. We note that the classifier is biased to overpredict speech from the target speaker, which is likely to be a problem for the baseline approach.

To use either of the proposed SU selection criteria from Sec. 3.4, one must choose a threshold value that tunes the selectivity of the method. As the threshold changes, the fraction of true and false positives (computed as a fraction of the total true or false selectable frames for each method) trace out a so-called ROC curve, which is graphed in Figure 3. The baseline, which is based on MAP classification and thus lacks a tuning parameter, is represented by a single dot in the same figure. It is seen that the baseline lies beneath and to the right of both curves, meaning that the proposed methods give strictly better performance for a number of threshold values.

4.3. Test-Set Results

Our final task is to compare the baseline and the proposed method on the held-out, test-set episodes. To ensure similar operating conditions, we tuned the proposed selection criteria to have the same validation-set true positive rate (70%) as the baseline. p_{worst} (threshold 0.84) gave the least false positives at this operating point.

Applied to the test set, the baseline extracted 1912 distinct segments (118 minutes) out of 656 minutes total. p_{worst} – chosen as our final proposed system – performed similarly, extracting 1969 segments (106 minutes of speech) out of 4331 automatically-identified

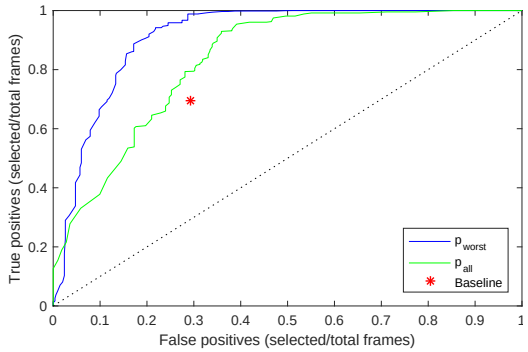

Fig. 3. ROC curves for the two segment-selection criteria in Sec. 3.4 and the baseline selector on the validation data.

Issue	Baseline	Proposed	p-value
None (problem-free)	70	217	$<10^{-44}$
No breath at the beginning	111	4	$<10^{-30}$
Backchannel from B present	37	17	$4.1{\cdot}10^{-3}$
Speech from B present	26	7	$6.4{\cdot}10^{-4}$
Noise	6	5	0.84

Table 4. Number of segments with different properties, out of 250 randomly sampled SUs extracted by either method on the test-set episodes, plus the p-value of a two-sided Barnard's test for each row.

target-speaker breath groups (318 minutes, i.e., 49% of all test audio). However, this does not mean that the methods are equal, since data quality is more important than quantity for high-quality TTS.

To assess the quality of the two automatically-extracted corpora in a meaningful way, we randomly sampled 250 segments extracted by each method and listened to them for any of a number of important issues that are likely to negatively affect TTS system training. In particular, we looked at whether segment starting points were appropriate and whether segments contained any non-target-speaker audio. Our findings are reported in Table 4. The proposed method is seen to substantially outperform the baseline both in terms of segmentation (appropriate boundaries) and selection (avoiding inappropriate audio). When it included non-target audio it was most often short backchannels. All improvements save for the reduction in noisy segments are statistically significant at the 0.05 level. This validates our proposal to account for breath in creating TTS corpora from found audio, and unlocks exciting new possibilities for conversational speech synthesis with large, natural datasets.

5. CONCLUSION AND FUTURE WORK

This paper considered the potential of breath-based methods in creating improved single-speaker spontaneous-speech TTS corpora from found conversational-speech audio such as podcasts, using only a small amount of coarsely annotated seed data. We found that image-based methods inspired by computational paralinguistics, supplemented by zero-crossing-rate information, provided good results in automatic annotation, and that the proposed approach leveraging breath information outperformed a baseline method based on directly selecting target speech without regard for breaths.

Our next step is to apply the method to a large number of podcast episodes, to produce a uniquely large conversational-speech corpus for TTS, and then to build speaking systems on this data, using them to improve spoken conversations between man and machine.

6. REFERENCES

[1] S. Andersson, J. Yamagishi, and R. A. J. Clark, "Synthesis and evaluation of conversational characteristics in HMM-based speech synthesis," *Speech Commun.*, vol. 54, no. 2, pp. 175–188, 2012.

[2] R. Dall, "Statistical parametric speech synthesis using conversational data and phenomena," Ph.D. dissertation, School of Informatics, The University of Edinburgh, Edinburgh, UK, 2017.

[3] T. Nagata, H. Mori, and T. Nose, "Dimensional paralinguistic information control based on multiple-regression HSMM for spontaneous dialogue speech synthesis with robust parameter estimation," *Speech Commun.*, vol. 88, pp. 137–148, 2017.

[4] É. Székely, J. Mendelson, and J. Gustafson, "Synthesising uncertainty: the interplay of vocal effort and hesitation disfluencies," *Proc. Interspeech 2017*, pp. 804–808, 2017.

[5] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," in *Proc. ICASSP*, 2018, pp. 5934–5938.

[6] J. Yamagishi, Z.-H. Ling, and S. King, "Robustness of HMM-based speech synthesis," in *Proc. Interspeech*, 2008, pp. 581–584.

[7] N. Braunschweiler and S. Buchholz, "Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality," in *Proc. Interspeech*, 2011, pp. 1821–1824.

[8] G. E. Henter, S. Ronanki, O. Watts, M. Wester, Z. Wu, and S. King, "Robust TTS duration modelling using DNNs," in *Proc. ICASSP*, vol. 41, 2016, pp. 5130–5134.

[9] J. Lorenzo-Trueba, F. Fang, X. Wang, I. Echizen, J. Yamagishi, and T. Kinnunen, "Can we steal your vocal identity from the Internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data," *arXiv preprint arXiv:1803.00860*, 2018.

[10] B. W. Schuller, Y. Zhang, and F. Weninger, "Three recent trends in paralinguistics on the way to omniscient machine intelligence," *J. Multimodal User In.*, vol. 12, no. 4, pp. 1–11, 2018.

[11] N. Braunschweiler and L. Chen, "Automatic detection of inhalation breath pauses for improved pause modelling in HMM-TTS," in *Proc. SSW*, vol. 8, 2013, pp. 1–6.

[12] S. H. Dumpala and K. N. R. K. Raju Alluri, "An algorithm for detection of breath sounds in spontaneous speech with application to speaker recognition," in *Proc. SPECOM*, 2017, pp. 98–108.

[13] T. Fukuda, O. Ichikawa, and M. Nishimura, "Detecting breathing sounds in realistic Japanese telephone conversations and its application to automatic speech recognition," *Speech Commun.*, vol. 98, pp. 95–103, 2018.

[14] B. Lei, S. A. Rahman, and I. Song, "Content-based classification of breath sound with enhanced features," *Neurocomputing*, vol. 141, pp. 139–147, 2014.

[15] A. G. Canal, M. Collery, V. Miloulis, and Z. Malisz, "Classification and clustering of clicks, breathing and silences within speech pauses," in *Proc. Laughter Workshop*, vol. 5, 2018, pp. 6–9.

[16] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[17] M. Włodarczak and M. Heldner, "Respiratory constraints in verbal and non-verbal communication," *Front. Psychol.*, vol. 8, no. 708, pp. 1–11, 2017.

[18] P. J. Price, M. Ostendorf, and C. W. Wightman, "Prosody and parsing," in *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod*, 1989, pp. 5–11.

[19] A. Rochet-Capellan and S. Fuchs, "Take a breath and take the turn: how breathing meets turns in spontaneous dialogue," *Phil. Trans. R. Soc. B*, vol. 369, no. 20130399, 2014.

[20] S. Amiriparian, N. Cummins, S. Julka, and B. W. Schuller, "Deep convolutional recurrent neural network for rare acoustic event detection," in *Proc. DAGA*, 2018, pp. 1522–1525.

[21] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. W. Schuller, "An image-based deep spectrum feature representation for the recognition of emotional speech," in *Proc. ACMMM*, 2017, pp. 478–484.

[22] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, "Snore sound classification using image-based deep spectrum features," in *Proc. Interspeech*, 2017, pp. 3512–3516.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[24] A. Stan, C. Valentini-Botinhao, B. Orza, and M. Giurgiu, "Blind speech segmentation using spectrogram image-based features and mel cepstral coefficients," in *Proc. SLT*, 2016, pp. 597–602.

[25] D. Ruinskiy and Y. Lavner, "An effective algorithm for automatic detection and exact demarcation of breath sounds in speech and song signals," *IEEE T. Audio Speech*, vol. 15, no. 3, pp. 838–850, 2007.

[26] K. Boakye, O. Vinyals, and G. Friedland, "Improved overlapped speech handling for speaker diarization," in *Proc. Interspeech*, 2011, pp. 941–944.