The CSTR entry to the Blizzard Challenge 2017

Srikanth Ronanki, Manuel Sam Ribeiro, Felipe Espic, Oliver Watts

The Centre for Speech Technology Research (CSTR), University of Edinburgh, UK

srikanth.ronanki@ed.ac.uk

Abstract

The annual Blizzard Challenge conducts side-by-side testing of a number of speech synthesis systems trained on a common set of speech data. Similar to 2016 Blizzard challenge, the task for this year is to train on expressively-read children's story-books, and to synthesise speech in the same domain. The Challenge therefore presents an opportunity to investigate the effectiveness of several techniques we have developed when applied to expressive and prosodically-varied audiobook data.

This paper describes the text-to-speech system entered by The Centre for Speech Technology Research into the 2017 Blizzard Challenge. The current system is a hybrid synthesis system which drives a unit selection synthesiser using the output from a neural network based acoustic and duration model. We assess the performance of our system by reporting the results from formal listening tests provided by the challenge.

Index Terms: Merlin, hybrid speech synthesis, unit selection, deep neural networks.

1. Introduction

The CSTR entry to this year's Blizzard Challenge builds on the hybrid Multisyn [1, 2] system submitted for last year [3]. Hybrid synthesis systems on the basis of target cost function [4, 5, 6, for example] employ statistical models to predict acoustic properties of speech thereby brings the benefits of extremely natural-sounding unit selection (which is unaffected by the degradations introduced by vocoding [7, 8]).

Similar to last year, the data used for this year's Challenge was obtained from professionally-read child-directed audio books and is therefore much more prosodically rich than the more standard prompt-based speech data. The experiment presented in [5, 6] established that improving the underlying SPSS of a hybrid synthesiser results in improvements to the concatenated output speech. Therefore, for our previous entry, we have incorporated two major improvements to the underlying SPSS model compared to the system presented in [6]: the decision tree duration model was replaced with a bi-directional long short-term memory (LSTM) recurrent neural network, and the feed-forward DNN acoustic model was replaced with an LSTM network.

Compared to our previous year entry, the text processing and speech parameterization steps are largely unchanged. Acoustic model prediction is slightly optimised for better prediction of fundamental frequency by adding supra-segmental features based on acoustic counts and is explained in 2.3.2. A notable exception is our attempt to smooth the joins: this new development is described in section 2.6 below. The neural networks used in this entry were trained using our open-source Merlin speech synthesis toolkit [9].

2. System Description

2.1. Data

The database – provided to the Challenge by Usborne Publishing Ltd. – consists of the speech and text of 56 children's audiobooks spoken by a British female speaker. We made use of a segmentation of the audiobooks carried out by two other Challenge participants[1][2] and kindly made available to other participants. The total duration of the audio is approximately 6 hours after segmentation. Three audiobooks from the given corpus were held out to act as an internal development set to gauge system performance before generating the final test data. The held-out data consists of three full short stories: *Goldilocks and the Three Bears*, *The Boy Who Cried Wolf* and *The Enormous Turnip*, having a total combined duration of approximately 10 minutes.

2.1.1. Sentence selection

For sentence selection, we have followed the same approach as last year. For clarity, we repeat the procedure followed from our previous year entry [3].

Harnessing the variety of speaking styles present in expressively-read audiobooks might enable us to produce less robotic-sounding TTS systems. However, initial experiments showed that the extreme variation in parts of the training data for the Challenge resulting in poor unit selection. We therefore filtered the data using the active learning approach described in [10]: 198 utterance-level acoustic features are extracted, and 15 sentences initially labelled as *keep* or *too expressive* by an expert listener. Uncertainty sampling [11] using an ensemble of decision trees was then used to select a further informative sample to be hand-labelled; this process continued for 20 minutes (real time). A classifier built on the entire set of hand-labelled data was then used to determine the subset of available sentences to be used for training. 20% of the training sentences were discarded in this way; informal comparison suggested this resulted in more stable synthesis with fewer unwarranted prosodic excursions.

2.2. Text processing

We have used Festival's English front-end with the British Received Pronunciation version of the Combilex lexicon [12]. 163 items were added to cover words appearing in the training data but otherwise absent from the dictionary. There were slight differences in the lexicon-lookup procedures used in preparing the annotation for training the SPSS model and those employed by the Festival front-end used for Multisyn. The resulting inconsistencies were dealt with by aligning the DNN's phone sequences to those expected by Multisyn in an ad hoc fashion and is similar to our previous year entry.

[1] Innoetics: https://www.innoetics.com

[2] IIIT-H: http://speech.iiit.ac.in

Word and syllable level vector representations were included, according to the method described in [13]. These were learned by taking counts of acoustic events of *f0* and energy stylized by clustered vectors and mean values defined over syllables or words. The training data available for the Challenge was used to learn these matrices. Experiments using vectors representations learned over a larger database of a different speaker, but we have observed that results were comparable with speaker-dependent vectors learned on a smaller database.

2.3. Parametric system

The parametric system was implemented using DNNs in a conventional two-stage approach. In the first stage, a duration model is used to predict phone durations to form frame-level linguistic features. In the second stage, an acoustic model is used to generate parameters from those linguistic features.

2.3.1. Duration model

The duration model trained for our entry to the challenge made use of a simple and straightforward approach with feed-forward neural networks (DNNs) as demonstrated in [14, 15]. The duration model is trained on the aligned data and generates state-level durations given phone-level linguistic features.

The described approach was used only to generated durations, which were then used to form frame-level linguistic features used as input in the generation of acoustic parameters. The hybrid Multisyn unit-selection system, however, does not make use of any duration-related features in its target cost function. Including such features in the unit selection process is left for future work.

2.3.2. Acoustic model

The linguistic features extracted from the front-end were converted to numerical vectors using a set of continuous and binary questions [9]. To these, we appended the syllable and word level vector representations based on acoustic counts [13]. The durations generated by the duration model described above were used to propagate all feature to frame-level. These frame-level feature vectors were then used as input to an acoustic model.

A feedforward neural network was trained at the frame-level to map linguistic inputs to vocoder parameters consisting of static and dynamic (delta and delta-delta) features. These acoustic parameters include 60 mel-cepstra coefficients, 25 band aperiodicities, log-f0, and a binary voicing decision. Maximum likelihood parameter generation (MLPG) and postfiltering are then applied to the generated acoustic parameters. In SPSS these parameter trajectories would then be passed through the vocoder to synthesize a speech waveform. Instead, we use them as targets for selecting waveform units.

Within each phone unit, generated parameters are split uniformly across time into 4 sections. A Gaussian distribution is then fitted for each sub-phone section of acoustic parameters. The variances of these Gaussian distributions are floored at 1% of the global variance per parameter [6].

The distributions associated with each of the 4 sub-phone sections are used to construct a diphone representation for the target utterance. To construct a diphone representation, we take the first or last 2 sections associated with its corresponding phones. Comparable distributions were generated for the diphone candidates in the unit database, based on vocoder parameters extracted from the training data and natural durations obtained by forced alignment.

2.3.3. Feature extraction

Phone sequences were obtained from the text using Festival [16]. Festvox's `ehmm` method [17] was used to modify the phone sequences by the insertion of acoustically-motivated pauses; A state-level forced alignment of these phone sequences with the sentence-segmented audio was then obtained using context-independent HMMs, similar to [18]. Each phone was then characterised by a vector of 481 text-derived binary and numerical features – a subset of the features used as decision-tree clustering questions in the HTS demo [19], adapted for our phoneset.

These questions included linguistic contexts such as quin-phone identity which are added at phone-level, and part-of-speech, positional information relating to syllables, words, phrases, *etc*. All numerical features are given as input (after appropriate normalisation) directly to the network, and not encoded as (for example) 1-of-K.

For duration modelling, all these features were used as input and normalised to the range of $[0.01, 0.99]$. The output for training is an five-dimensional vector of durations for every phone, comprising five sub-state durations.

For acoustic modelling, the input uses the same features as duration prediction, to which 9 numerical features were appended. These capture frame position in the HMM state and phoneme, state position in phoneme, and state and phoneme duration, similar to [18].

The speech data was analysed with STRAIGHT [20], and each 5ms frame was represented using 60 mel cepstral coefficients (MCC), measures of aperiodicity in 25 frequency bands (BAP), logarithmic F_0 interpolated through unvoiced regions, and a binary voicing feature. These 87 static features were supplemented with delta and delta-delta features, and for both the duration and acoustic data, a per-component mean and variance normalisation was applied prior to model training, with the transformation reversed as part of synthesis.

2.3.4. Duration and acoustic model training

For the duration model, we have used 481-dimensional binary and continuously valued feature vectors as input. Its output was a 5-dimensional feature vector representing state durations in terms of frames. The model was defined to be 6 feedforward hidden layers, each with 1024 nodes, using the *tanh* activation function. Mini batch size was set to 64 and learning rate was set to 0.002, being was reduced by 50% with each epoch after the first 10 training epochs.

For the acoustic model, we have used the same 481-dimensional feature vector representing linguistic features. To these, we added syllable and word level vector representations spanning a window of 3 units. Nine frame-level features were included according to [18] and available from [9]. The input vector to the acoustic models consisted of a total of 1900 dimensions. The model consisted of 6 feedforward hidden layers, each with 1024 nodes, using the *tanh* activation function. Mini-batch was set to 256 and remaining parameters were identical to the duration model.

2.4. Unit selection waveform renderer

For unit selection, we have followed the same approach as last year. For clarity, we repeat the procedure followed from our previous year entry [3].

A modified form of Festival's Multisyn engine [2] was used for the unit selection stage of our system. To compare the suit-

ability of a given candidate diphone in the unit database with the 4 distributions representing a synthesised diphone, the symmetrised Kullback Leibler divergence (KLD) [21] is used. The KLD is computed between each of the 4 candidate unit's distributions and the corresponding target unit distributions individually. The resulting 4 scores are then summed to produce the final target score.

The standard Multisyn join cost (sum of distances between 12 MFCCs, f_0 and energy from the frame either side of the join) is retained, as well as the standard pre-selection criterion of candidate units (by matching diphone identity). The standard Multisyn Viterbi search (with pruning to reduce the search time) is performed in order to optimise target cost and join cost. Also the standard Multisyn back-off rules are used where the target diphone to be synthesised is not present in the training data.

2.5. Speech synthesis

At synthesis time, duration is predicted first, and is used as an input to the acoustic model to predict the speech parameters. Maximum likelihood parameter generation (MLPG) [22] using variances computed from the training data was applied to the output features for synthesis, and spectral enhancement post-filtering was applied to the resulting MCC trajectories. These parameter trajectories are then used to produce diphone coefficients. The Festival Multisyn engine was used to compute the target and joint cost between target unit and pre-selected candidate units to select the final candidate, as explained above. The final waveform synthesis was done by joining the selected units. An additional smoothing and post-modification of prosody was performed during joining the units and is explained in below section.

2.6. Concatenation and join smoothing

The selected waveform units are parameterised by using the method proposed in [23]. It extracts pitch synchronous speech features in a frame-by-frame basis, describing the complex spectra and F0 contour. The correction/smoothing operations are performed over these features to produce seamless concatenation of units.

2.6.1. Concatenation and correction of F0 contours

The $F0$ mid point ($F0_m$) between two consecutive units is given by $F0_m = (F0_p[N_p - 1] + F0_c[0])/2$, where p means preceding unit, c current unit, and N is the unit length in frames.

Then, the slope of the F0 contours of both units are adjusted to reach the $F0_m$ just in the join location. The corrected $F0$ contours are computed by the Equations 1 and 2.

$$F0'_c[n_c] = F0_c[n_c] + (F0_m - F0_c[0]) \cdot \left(\frac{n_c}{1 - N_c} + 1 \right) \quad (1)$$

$$F0'_p[n_p] = F0_p[n_p] + (F0_m - F0_p[N_p - 1]) \cdot \frac{n_p}{N_p - 1} \quad (2)$$

Where $F0'$ is the corrected $F0$, and n is the frame index within each unit. After having all the corrected $F0$ contours for all the units, these are appended building a single $F0$ contour for the whole sentence.

2.6.2. Spectral concatenation and smoothing

Basically, it is done by overlapping and crossfading the complex FFT spectra of two consecutive units. Some extra frames are extracted from the sources, so the units can be overlapped without affecting their expected locations in the synthesised waveform. Three extra frames on each side of the units are extracted from the sources, thus an overlap of seven frames around the joins is produced.

The FFT complex spectrum S is derived from the parameters proposed in [23], M, R, and I, by $S = M \cdot (R + Ij)$. The crossfade is linearly applied to mix the FFT complex spectra of two consecutive units, progressively. It is seven frames length, and in case that a unit is too short, the crossfade is shortened accordingly.

After performing this operation on every join, the FFT complex spectra of all the units are concatenated producing a single complex spectra stream, that describes the whole utterance.

Finally, the signal is synthesised by converting the FFT complex spectra to time domain, and applying Pitch Synchronous Overlap-Add as explained in [23], using the corrected $F0'$ contour.

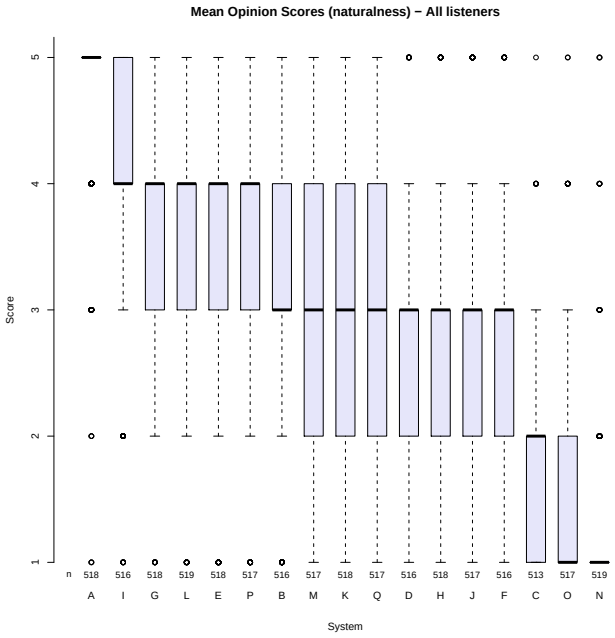

Figure 1: *Our system(E): Mean opinion score for naturalness of the synthesized speech with ratings from all listeners.*

2.7. Paragraph-level synthesis

From the sentences synthesised in this way, files were made containing whole paragraphs, chapters and books as required by the Challenge by simply concatenating the waveforms. While proper exploitation of long-distance contexts ought to improve synthesis quality, no contexts outside the current sentence were used for the present submission.

3. Results

The identifier for our system in the published results is E.

3.1. Naturalness

Mean opinion scores for naturalness from all listeners on book sentences are shown in Figure 1. In our discussion, we make use of the published statistical analysis of the results at 1% level with Bonferoni corrected alpha) [24]. Our system outperformed all three baselines (systems B, C and D). Among the 12 other challenge participants, our system is outperformed only by a single system (I). The same trend can be seen across the scores

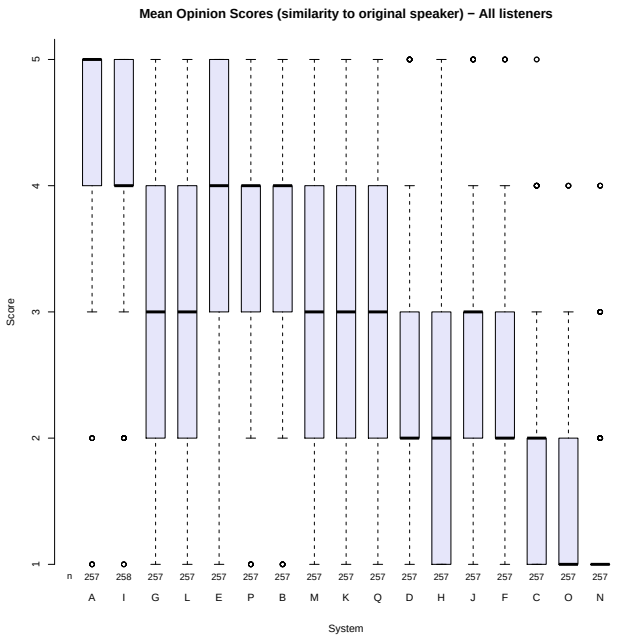

Figure 2: *Our system(E): Mean opinion score for speaker similarity with ratings from all listeners.*

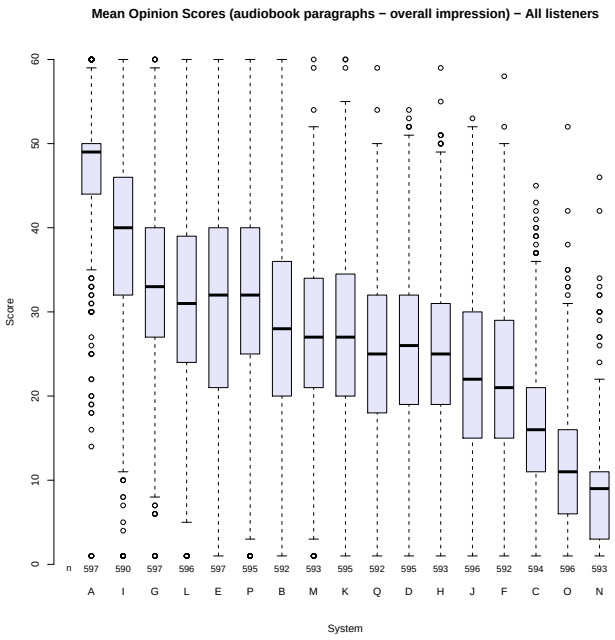

Figure 3: *Our system(E): Mean opinion score for overall impression with ratings from all listeners.*

made by paid listeners, on-line volunteers and considering ratings only from speech experts, no other system was significantly better than ours. Overall, our system outperformed 11 out of 15 other systems (including three baselines) evaluated for listening test.

3.2. Speaker similarity

The mean opinion scores for speaker similarity from all listeners on book sentences are shown in Figure 2. Considering ratings from all listeners (or any other listener group), no other system was significantly better than ours and our system was in turn significantly better than 13 other systems. These results show the effectiveness of waveform concatenation systems for speaker similarity.

3.3. Evaluation of audiobook paragraphs

We now consider the results for evaluation of audiobook paragraphs – that have been evaluated on several other factors like stress, intonation, emotion, pleasantness, listening effort, speech pauses and overall impression. Considering ratings from all listeners on overall impression, our system showed similar performance as in the case of the isolated sentence evaluation of naturalness and speaker similarity. Only one system (I) outperformed us and our system was significantly better in turn than 11 other systems (cf. Figure 3). A similar trend can be seen across the scores made by speech experts, online volunteers and paid listeners. Considering ratings for other individual factors (e.g., intonation, emotion and pleasantness) from all listeners, again only system I consistently outperformed ours. Overall, our system outperforms between 7 and 11 other systems in evaluation of each of these factors, performing best in emotion and pleasantness.

3.4. Intelligibility (SUS)

We now consider the results for intelligibility of semantically unpredictable sentences (making use of the published statistical analysis of significant difference between word error rates of the systems). Taking into account ratings from all listeners, there are only three other systems out of 15 (D, L, and M) significantly better than ours. Considering only paid listeners, there are only two other systems (D and L) significantly better than ours. Out of 15 other systems evaluated by paid listeners, 10 were not significantly more or less intelligible than ours, 3 were significantly less intelligible, and only 2 significantly more intelligible. The results show that our system is quite effective on intelligibility as well. Overall, our system has shown consistent performance (standing in the top four) in all the factors evaluated for the Challenge.

4. Conclusions and future work

For this year's CSTR Blizzard Challenge entry, the hybrid system submitted for last year [3] was slightly optimized in acoustic modeling for better prediction of F0 and performed smoothing between joins.

The results of the evaluation are on the whole very positive, but there are still a number of potential future improvements which could be made to the hybrid synthesis system described here. These include adopting consistent lexicon-lookup for both the SPSS and unit selection systems, making use of same acoustic features for both join and target cost, prediction of phrase breaks, and the explicit inclusion of predicted duration in the unit selection synthesis target cost.

Reproducibility: We used the Open Source Merlin toolkit[3] for parameter prediction and Festival Multisyn[4] for unit-selection.

Acknowledgement: Watts was supported in this research by EPSRC Standard Research Grant EP/P011586/1, *Speech Synthesis for Spoken Content Production (SCRIPT)*.

[3] https://github.com/CSTR-Edinburgh/merlin
[4] http://www.cstr.ed.ac.uk/projects/festival

5. References

[1] R. A. Clark, K. Richmond, and S. King, "Festival 2–build your own general purpose unit selection speech synthesiser," in *Proc. SSW*, 2004.

[2] R. A. Clark, k. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[3] T. Merritt, S. Ronanki, Z. Wu, and O. Watts, "The CSTR entry to the Blizzard Challenge 2016," in *Proc. Blizzard Challenge workshop*, 2016.

[4] Z.-J. Yan, Y. Qian, and F. K. Soong, "Rich-context unit selection (rus) approach to high quality tts," in *Proc. ICASSP*, 2010, pp. 4798–4801.

[5] Y. Qian, F. K. Soong, and Z.-J. Yan, "A unified trajectory tiling approach to high quality speech rendering," *IEEE Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 280–290, 2013.

[6] T. Merritt, R. A. J. Clark, Z. Wu, J. Yamagishi, and S. King, "Deep neural network-guided unit selection synthesis," in *Proc. ICASSP*, 2016.

[7] H. Zen, A. Senior, and M. Schuster, "Statistical Parametric Speech Synthesis Using Deep Neural Networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[8] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, "Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech," in *Proc. Interspeech*, 2014, pp. 1504–1508.

[9] Z. Wu, O. Watts, and S. King, "Merlin: An Open Source Neural Network Speech Synthesis System," in *Proc. SSW*, Sunnyvale, USA, 2016.

[10] O. Watts, A. Stan, R. Clark, Y. Mamiya, M. Giurgiu, J. Yamagishi, and S. King, "Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis," in *8th ISCA Workshop on Speech Synthesis*, Barcelona, Spain, Aug. 2013, pp. 121–126.

[11] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994, pp. 3–12.

[12] S. Fitt and K. Richmond, "Redundancy and productivity in the speech technology lexicon - can we do better?" in *Proc. Interspeech 2006*, Sep. 2006.

[13] M. S. Ribeiro, O. Watts, and J. Yamagishi, "Learning word vector representations based on acoustic counts," in *Proc. Interspeech*, Stockholm, Sweden, August 2017.

[14] S. Ronanki, Z. Wu, O. Watts, and S. King, "A Demonstration of the Merlin Open Source Neural Network Speech Synthesis System," in *Proc. SSW*, Sunnyvale, USA, 2016.

[15] S. Ronanki, O. Watts, S. King, and G. E. Henter, "Median-based generation of synthetic speech durations using a non-parametric approach," in *IEEE workshop on Spoken Language Technology*, San Diego, California, 2016.

[16] A. Black, P. Taylor, R. Caley, R. Clark, K. Richmond, S. King, V. Strom, and H. Zen, "The festival speech synthesis system, version 1.4. 2," *Unpublished document available via http://www. cstr. ed. ac. uk/projects/festival. html*, 2001.

[17] K. Prahallad, A. W. Black, and R. Mosur, "Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis," in *Proc. ICASSP*, 2006, pp. I–853–I–856.

[18] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*, 2015, pp. 4460–4464.

[19] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. SSW*, vol. 6, 2007, pp. 294–299.

[20] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoust. Sci. Technol.*, vol. 27, no. 6, pp. 349–353, 2006.

[21] J. R. Hershey and P. A. Olsen, "Approximating the Kullback-Leibler divergence between Gaussian mixture models," in *Proc. ICASSP*, 2007.

[22] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, vol. 3, 2000, pp. 1315–1318.

[23] F. Espic, C. Valentini-Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Interspeech*, Stochohlm, Sweden, August 2017.

[24] R. A. Clark, M. Podsiadlo, M. Fraser, C. Mayo, and S. King, "Statistical analysis of the Blizzard Challenge 2007 listening test results," *Proc. Blizzard Challenge Workshop*, 2007.