Sentence-level control vectors for deep neural network speech synthesis

Oliver Watts, Zhizheng Wu, Simon King

Centre for Speech Technology Research, University of Edinburgh, UK

owatts@inf.ed.ac.uk, zhizheng.wu@ed.ac.uk, Simon.King@ed.ac.uk

Abstract

This paper describes the use of a low-dimensional vector representation of sentence acoustics to control the output of a feed-forward deep neural network text-to-speech system on a sentence-by-sentence basis. Vector representations for sentences in the training corpus are learned during network training along with other parameters of the model. Although the network is trained on a frame-by-frame basis, the standard frame-level inputs representing linguistic features are supplemented by features from a projection layer which outputs a learned representation of sentence-level acoustic characteristics. The projection layer contains dedicated parameters for each sentence in the training data which are optimised jointly with the standard network weights. Sentence-specific parameters are optimised on all frames of the relevant sentence – these parameters therefore allow the network to account for sentence-level variation in the data which is not predictable from the standard linguistic inputs. Results show that the global prosodic characteristics of synthetic speech can be controlled simply and robustly at run time by supplementing basic linguistic features with sentence-level control vectors which are novel but designed to be consistent with those observed in the training corpus.

Index Terms: text-to-speech, speech synthesis, controllable speech synthesis, audiobooks, deep neural nets, neural net embeddings, unsupervised learning

1. Introduction

Conventional data-driven text-to-speech (TTS) generally aims to achieve adequate neutral prosody. This may be acceptable for synthesising isolated sentences, but for whole paragraphs or stories such repetitive neutral prosody – unvarying between sentences – is fatiguing for listeners and unpleasant to listen to. There has been much recent interest in training TTS systems on speech from audiobooks [1, 2]. When an expert voice talent makes such recordings, they modulate intonation, rhythm and intensity of their speech from sentence to sentence in order to convey the coherence of the text and engage listeners' attention. With growing interest in training systems on such data – and synthesising speech in the same domain – it is important to establish effective techniques for modelling and controlling prosodic variation above the sentence level.

For HMM-based synthesis, several techniques have been proposed that can be controlled externally with exogenous variables, such as cluster-adaptive training (CAT) [3], multiple-regression hidden semi-Markov models (HSMM) [4] and eigenvoices [5]. Note that the specific tasks performed by control vectors differ (approximation of speaker characteristics, speaking style, emotion, etc.) but these models all have in common the possibility for external control. Speech synthesis with deep neural networks (DNN) has recently been shown to be competitive in quality with that of HSMM-based systems [6, 7, 8], however, and so it is desirable to find equivalent techniques for the exogenous control of DNN-based systems. We here experiment with a means of 'steering' an otherwise conventional DNN TTS system at the sentence level using exogenous control vectors (CVs). Unlike the previously cited work on CAT, MRHSMM and eigenvoices, we train our models from a 'flat start' with randomly initialised CVs for the training data. We thus learn in an unsupervised manner a space of sentences which captures the dimensions of variation in the training data, and which can then be used to modulate the characteristics of synthetic speech on a sentence-by-sentence basis.

We present several systems, each of which is the result of different training and synthesis time configurations. Analysis of the systems' output is presented, along with the results of an evaluation using randomly sampled CVs, and 'oracle' CVs inferred from held-out test audio. The evaluation of randomly-sampled CVs tests the hypothesis that *any* reasonable variation from sentence to sentence is preferable to conventional monotonous prosody, even if that variation is unconnected to the text. The evaluation of 'oracle' CVs is designed to test whether optimal low-dimensional CVs are adequate to capture sentence-level variation in speech.

2. DNN with sentence-level control vectors

2.1. Basic DNN

The bulk of the work of synthesis in all our systems is performed by a conventional DNN TTS model similar to the baseline system in [8]. This is shown by the unshaded parts of Figure 1: a feed-forward multilayer perceptron with multiple hidden layers, whose inputs are numerical representations of conventional linguistic features coded at frame level. Each hidden layer computes a representation of the previous (output or hidden) layer as a non-linear function of the previous layer's representation. The network's output is computed as a linear function of the final hidden layer, and is a frame of parameters which can – directly or in some smoothed form – be used to drive a vocoder.

2.2. Control vectors

The novel part of the systems is shown by the shaded parts of Figure 1. This part of the model supplements the standard frame-level inputs with features from a projection layer which outputs a learned representation of sentence-level acoustic characteristics of the current sentence. The projection layer's input consists of an n-dimensional binary vector, where n is the number of sentence tokens in the training corpus and where 1 bit of the vector is turned on to indicate the index of the current sentence token. Its (linear) output is a d-dimensional vector, where d is the dimensionality of the CV space used by the system. The projection layer's parameters are represented as an $n \times d$ matrix $\boldsymbol{P}$, each of whose rows is dedicated to a sentence of the training

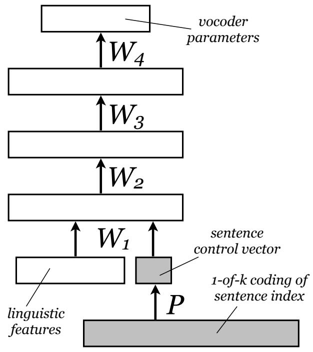

Figure 1: Architecture of networks trained

data. $\boldsymbol{P}$ is optimised jointly with the weights and biases of the network's k hidden layers ($\boldsymbol{W}_{1...k}$). Although the network is trained on a frame-by-frame basis, the sentence-specific parameters are constrained to be the same for all frames of the relevant sentence – these parameters therefore allow the network to account for sentence-level variation in the data which is not predictable from the standard linguistic inputs. A sentence-level representation of the training data emerges which minimises the loss function used to train the network as a whole.

2.3. Synthesis methods

Given a trained model, CVs need to be supplied at run-time. Various options for obtaining these exist: one possibility is to use the learned sentence space as an interface for allowing a human operator to control the system's output. Full investigation of this possibility is out of the scope here, but it has encouraged our focus on 2-dimensional sentence vectors which could be controlled manually. Ultimately we are interested in predicting control vectors from text. Again, however, we leave such prediction with an external model for future work. Instead, the main hypothesis that the current work tests is that any reasonable variation from sentence to sentence is preferable to doing the same thing on each sentence as conventional models do, even if that variation is not conditioned on the text input. One system therefore randomly samples CVs at synthesis time. Finally, there are two methods of synthesis which we regard as baseline and topline: using the mean of the training vectors as a fixed CV which remains unchanged from one sentence to the next, and using 'oracle' CVs for the test set. These oracle CVs are inferred by optimisation on the test set speech.

2.4. Previous work

The use of *extended backpropagation* to learn representations of network inputs as weights on connections feeding into the input layer was described in [9]. The use of such projection layers to represent multiple words or other textual units in a context in a way that is invariant to their position in that context has become widespread in language modelling [10, 11] as well as being applied to other tasks (e.g. letter-to-sound [12]; phrase-break prediction [13]). The idea is used for acoustic modelling in [14, 15, 16], where the CVs are speaker codes for rapidly adapting DNN-based speech recognition system to new speakers. Codes for new speakers are inferred from adaptation data in a way similar to that in which we obtain our 'oracle' CVs.

As mentioned, our method bears some resemblance to CAT and MRHSMM for HMM-based systems, in that these methods allow control of a synthesiser by means of an external control vector. In contrast to [3] and [17], however, we initialise our utterance representations randomly, and learn them in an entirely unsupervised fashion.

Table 1: *Summary of systems built.*

Synthesis method	CV dimensions: 2	5	10
Fixed	F	F_5	F_{10}
Sampled	S	–	–
Oracle	O	O_5	O_{10}

3. Experiments

3.1. Systems built

Table 1 summarises the systems built for the objective and subjective evaluation and informal analysis presented here. The only hyperparameter varied between systems is the dimensionality of the sentence CVs (columns of Table 1). We are particularly interested in the 2-dimensional case as it of particular relevance to human-controllable speech synthesis, and the listening tests evaluate only systems with 2-dimensional vectors. The rows of Table 1 indicate the different procedures for obtaining CVs at synthesis time, mentioned in Section 2.3, and which will be explained in detail in Section 3.5. The data used for training the systems will now be briefly outlined, as well as the methods used to train and assemble their front- and back-ends.

3.2. Data

The speech database for the pilot task of the 2015 Blizzard Challenge was used for these experiments [18]. The database – provided to the Challenge by Usborne Publishing Ltd. – consists of the speech and text of 22 children's audiobooks spoken by a British female speaker; the considerable prosodic variation of the corpus makes it ideal for testing techniques for global prosody control. The total duration of the audio is approx. 2 hours; for the purposes of this paper, 10% of the data was set aside as a test set and consists of three whole short stories: *Goldilocks and the Three Bears*, *The Boy Who Cried Wolf* and *The Enormous Turnip*, with a total duration of approximately 12 minutes.

The segmentation of the data distributed for the Challenge does not always divide the text and audio into whole sentences, and the time-aligned transcript has been lowercased and stripped of all punctuation. The original running text of the audiobooks with punctuation and case information intact was included as part of the release, however, and before any voices were built, a segmentation and transcription of the data respecting sentence boundaries and containing full punctuation were obtained by merging the running texts and unpunctuated time-aligned transcripts semi-automatically.

The rechunked data consists of 1995 and 239 sentences for train and test sets. The underlying sampling rate of the lossy-coded speech data distributed for the challenge was 44.1 kHz; it was downsampled to 16kHz for our experiments. Speech parameters were extracted from the downsampled speech using the GlottHMM vocoder [19]. Source and filter separation was achieved using glottal inverse filtering of the speech waveform; 30 line spectral frequency (LSF) coefficients representing vocal tract shape were extracted, along with several sets of parameters to characterise the estimated glottal source: 10 voice source LSF coefficients, the harmonic-to-noise ratio (HNR) in five frequency bands, energy, and fundamental frequency (F_0). Velocity and acceleration coefficients were computed for all aforementioned parameters and appended to the feature vector.

3.3. Front-end

Text-normalisation is performed in our front-end by a rule-based module which depends on long lists of acronyms, abbreviations, etc. Part-of-speech tags are assigned to words with a maximum entropy tagger [20] released publicly already trained [21]. Phonetic forms of words are looked up in a British English received pronunciation lexicon derived from the Combilex lexicon [22], chosen as a good match for the reader's accent. A letter-to-sound predictor based on joint multigrams [23] was trained on this lexicon to handle out-of-vocabulary words. Phonetic features such as place and manner of articulation are obtained for each phone by table lookup.

An HMM-based aligner was trained from a flat start on the data in order to determine the start and end points of each segment in the data. The whole state-alignment is retained and added to the annotation. The model allows silence to be inserted between words; a duration threshold (50ms) is used to flag short silences as spurious, which are then discarded. The retained silences are treated as phrase-boundaries. The aligner is retained to be used to force-align the test set for TTS to obtain natural durations and phrasing. After the positions of silences have been determined, several post-lexical rules (including e.g. handling British English linking-*r*) are applied.

From the corpus annotation described, frame-level linguistic feature files were prepared. These contain c.600 values per frame, and code similar features to those described in [6]. Phonetic and part of speech features are encoded as 1-of-k subvectors, and position and size information (including position of frame in current state and state in current phone) are encoded with continuous values. The features derived from (oracle) durations described in [6] were not used as these were found to unfairly improve performance, due to correlation of variations in segment duration with e.g. the presence of F_0 excursions. Features characterising the sentence by its length were excluded: the sentence CVs should remove the need for these.

3.4. Acoustic model training

For DNN training, 95% of the frames labelled as silence were removed. The unvoiced regions of the F_0 track were interpolated, and voicing was represented in a separate stream. Linguistic input features were normalised to the range [0.01, 0.99] and acoustic features standardised.

All systems had 6 hidden layers, each of 1024 units with tanh hidden unit non-linearities, and a linear activation function at the output layer. Network parameters (hidden layer weights/biases, output layer weights/biases, projection layer weights) were initialised with small non-zero values, and the network was optimised from this flat start with stochastic gradient descent to minimise the mean squared error between its predictions and the known acoustic features of the training set. L_2 regularisation was applied to the hidden layer weights with a penalty factor of 0.00001. Mini-batches consisted of 256 frames. For the first 15 epochs, a fixed learning rate of 0.002 was used with a momentum of 0.3. After 15 epochs, the momentum was increased to 0.9 and from that point on the learning rate was halved after each epoch. The learning rate used for the top two layers was half that used for the other layers.

5% of the training utterances were held-out from training for validation purposes; after each epoch, sentence CVs for

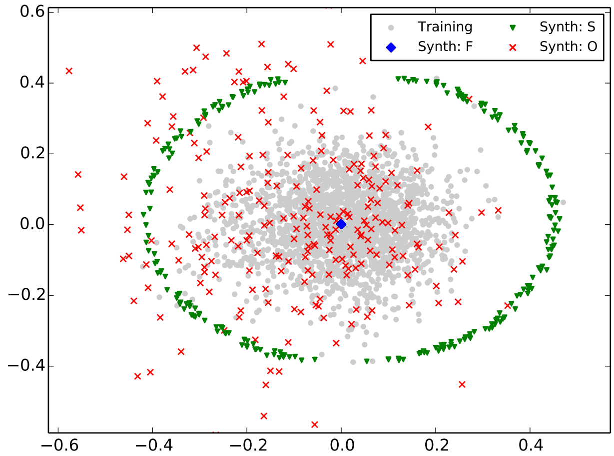

Figure 2: CVs learned in training and used at synthesis time

these held out frames were updated by doing stochastic gradient descent in the same way as for the training set, but updating only the relevant projection layer weights. Then network performance was evaluated by passing the development data forwards through the network and computing the loss function. Training finished when performance on the validation set stopped improving. Training took 32, 40 and 33 epochs on the systems employing 2-, 5- and 10-dimensional CVs respectively.

3.5. Speech synthesis

The aligner created during front-end training was used to impose natural state durations, pause positions and phrasing on the annotation of the test set.

CVs were made in different ways for systems on each row of Table 1. Training and test set CVs for systems with 2-dimensional CVs are shown in Figure 2. Systems F, F_5, and F_{10} all used the mean vectors of the CVs learned during training. For system S, CVs were sampled from the sentence space; it was found that sampling from a normal distribution fitted to the training CVs gave speech that was not much more varied than that of system F. To avoid the dominance of typical values near the mean whilst at the same time avoiding the generation of extreme outlying values, CVs were uniformly sampled between 3.8 and 4.0 standard deviations from the mean of a diagonal covariance Gaussian fitted to the CVs learned for training sentences. Finally, oracle CVs for systems O, O_5, and O_{10} were inferred from the audio of the test set; stochastic gradient descent was performed on the test set until convergence, updating only rows of matrix $\boldsymbol{P}$ dedicated exclusively to modelling the test data – other network parameters were left unchanged.

Figure 2 shows the control vectors learned in training and used at synthesis time by the systems using 2-dimensional control vectors (F, S and O). It can be seen that test set CVs for both the O-systems and system S are more extremely distributed than we might expect would be appropriate from the distribution of training CVs. In the case of system S, the sampling distribution was manually set via informal listening, and relatively extreme values were chosen. This is consistent with previous experience in controllable speech synthesis: [24] notes that to properly steer a data-driven articulatory-controllable to produce modified vowels, tongue movements must be specified of a far greater magnitude than those observed in the training data.

After CVs were determined for test utterances, labels were

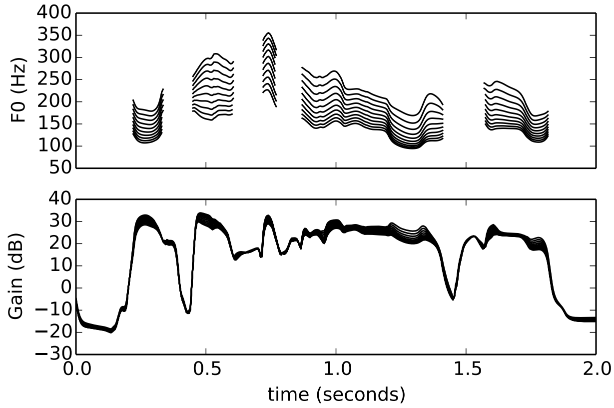

Figure 3: Variation in synthetic F_0 and gain for a single utterance fragment as CVs are manipulated over 10 repetitions

created for the test set. As predictions of the acoustic values for neighbouring frames are made independently, a parameter generation algorithm developed for HMM-based speech synthesis [25] is used with pre-computed variances from the training data to obtain smooth and speech-like vocoder parameter trajectories from the de-standardised DNN output features. The resulting trajectories for the LSF stream were enhanced by imposing the global variance of the training data using the simple z-score transform approach suggested by [26]. A modified form was used: best results were obtained by interpolating global variance and synthesised sentence variance with even weights.

3.6. Objective evaluation and analysis

Objective evaluations indicates that higher CV dimensionality improves prediction performance, as does using oracle CVs. Full details are omitted for reasons of space.

To get an informal impression of the meaning of the dimensions of the sentence space, we synthesised 100 repetitions of a few sentences whilst manually manipulating the values of the 2-dimensional CV. We chose 100 points evenly spread across the rectangle delimited by the minimum and maximum values along each axis of training set CVs.[1] The main dimension of variation is from the bottom left of Figure 2 to top right. Figure 3 shows synthetic F_0 and gain for a 10 repetitions of a single utterance fragment (*'Who's been sitting in my chair?'*), with CVs spaced evenly along this diagonal, starting at approximately coordinates (-0.4, -0.4) in Figure 2 and ending at approximately (0.5, 0.4). Absolute mean F_0 and gain both increase as we move the CV along this diagonal; however, the changes are much more complex and subtle than a global shift in values. Note how the F_0 contour on the word *been* (around 0.5 seconds) has an inflection which is inverted from the lower to the higher samples; in some places variation in F_0 increases more than in others. The sentence space allows us to alter global characteristics whilst respecting correlations between parameters.

3.7. Subjective evaluation

The 240 sentences from the 3 stories of the test set synthesised as described in Section 3.5 were concatenated back into 70 chunks of audio corresponding to book pages for the evaluation [27]. This is because the listening test is designed to test the effect on listeners of between-sentence variation. Each page on average contains 3.4 sentences and lasts 10.3 seconds.

Table 2: *Subjective results.*

List. ID	S > F (%)	List. ID	O > F (%)
1	47.14	11	40.00
2	35.71	12	61.43
3	47.14	13	55.71
4	37.14	14	54.29
5	42.86	15	54.29
6	47.14	16	50.00
7	51.43	17	51.43
8	50.00	18	50.00
9	48.57	19	50.00
10	51.43	20	52.86
All	45.86	All	52.00

Two tests were conducted: one comparing systems F and S, and the other comparing F and O. For each test, 10 paid native speakers of English were asked to listen to 70 pairs of stimuli and asked to say which they preferred. Specifically, they were asked to 'choose the version which you would prefer to hear if you were listening to stories like this for fun'. In each of the 70 pairs, the same page text was synthesised by the two different systems under evaluation. The ordering of the pairs was fixed to the page-order for the original stories, but the order of systems within each pair was balanced and randomised separately per listener. The listening test was conducted in purpose-built listening booths using high-quality headphones. Different listeners were employed for each evaluation.

Results are shown in Table 2. Results for pooled listeners (bottom row) force us to reject our hypothesis that random variation between sentences is better than fixed prosody (at least in the form that we realised the variation): there is a preference for system F over S which a binomial test indicates is significantly different from chance ($\alpha = 0.05$).

Results of the test comparing O and F show no significant difference when listeners' results are pooled. However, there is only a single listener (11) who prefers O's samples less than half the time; all others either prefer the oracle system O (listeners 12–15) or have no preference (16–20).

4. Conclusions

We have shown how the global prosodic characteristics of synthetic speech can be controlled simply and robustly at run time by supplementing basic linguistic features with sentence-level control vectors. Our results indicate that listeners have mixed reactions to prosodically more varied speech even when controlled by oracle CVs, which in itself is a motivation for making TTS more controllable. The hypothesis that 'any variation is better than no variation' was rejected: care needs to be taken that the variation is appropriate for the text being synthesised, which provides motivation for our ongoing work on learning to predict control vectors from text.

5. Acknowledgement

This research was supported by EPSRC Programme Grant EP/I031022/1, Natural Speech Technology (NST). The NST research data collection may be accessed at `http://datashare.is.ed.ac.uk/handle/10283/786`

[1] These samples can be heard at `http://homepages.inf.ed.ac.uk/owatts/papers/IS2015_sentence_control/`

6. References

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

[2] É. Székely, T. G. Csapó, B. Tóth, P. Mihajlik, and J. Carson-Berndsen, "Synthesizing expressive speech from amateur audiobook recordings," in *Proc. Spoken Language Technology Workshop*, 2012, pp. 297–302.

[3] L. Chen, N. Braunschweiler, and M. Gales, "Speaker and expression factorization for audiobook data: Expressiveness and transplantation," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 4, pp. 605–618, April 2015.

[4] T. Nose, M. Tachibana, and T. Kobayashi, "HMM-Based Style Control for Expressive Speech Synthesis with Arbitrary Speaker's Voice Using Model Adaptation," *IEICE Transactions*, vol. 92-D, no. 3, pp. 489–497, 2009.

[5] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-based speech synthesis," in *7th International Conference on Spoken Language Processing*, J. H. L. Hansen and B. L. Pellom, Eds. ISCA, 2002.

[6] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 7962–7966.

[7] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural network (DNN) for parametric TTS synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. IEEE, 2014, pp. 3829–3833.

[8] "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[9] R. Miikkulainen and M. G. Dyer, "Forming global representations with extended backpropagation," in *Proceedings of the IEEE International Conference on Neural Networks* (San Diego, CA), vol. I. Piscataway, NJ: IEEE, 1988, pp. 285–292.

[10] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[11] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492 – 518, 2007.

[12] K. J. Jensen and S. Riis, "Self-organizing letter code-book for text-to-phoneme neural network model," in *INTERSPEECH*, 2000, pp. 318–321.

[13] O. Watts, S. Gangireddy, J. Yamagishi, S. King, S. Renals, A. Stan, and M. Giurgiu, "Neural net word representations for phrase-break prediction without a part of speech tagger," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 2618–2622.

[14] J. S. Bridle and S. J. Cox, "RecNorm: Simultaneous Normalisation and Classification applied to Speech Recognition," in *Advances in Neural Information Processing Systems 3*, R. Lippmann, J. Moody, and D. Touretzky, Eds., 1991, pp. 234–240.

[15] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7942–7946.

[16] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1713–1725, Dec 2014.

[17] T. Nose, J. Yamagishi, and T. Kobayashi, "A style control technique for HMM-based expressive speech synthesis," *IEICE Trans. Information and Systems*, vol. E90-D, no. 9, pp. 1406–1413, September 2007.

[18] CSTR, "Usborne Publishing Ltd release of audiobook recordings for Blizzard 2015," http://www.cstr.ed.ac.uk/projects/blizzard/2015/usborne_blizzard2015/, 2015, [Online; accessed 10-June-2015].

[19] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 153–165, 2011.

[20] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

[21] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.

[22] K. Richmond, R. Clark, and S. Fitt, "On generating Combilex pronunciations via morphological analysis," in *Proc. Interspeech*, Makuhari, Japan, September 2010, pp. 1974–1977.

[23] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434 – 451, 2008.

[24] Z. Ling, K. Richmond, and J. Yamagishi, "Articulatory control of HMM-based parametric speech synthesis using feature-space-switched multiple regression," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 207–219, 2013.

[25] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, vol. 3, 2000, pp. 1315–1318 vol.3.

[26] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, "Ways to implement global variance in statistical speech synthesis," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 1436–1439.

[27] O. Watts, "Listening test materials for 'Sentence-level control vectors for deep neural network speech synthesis'," http://www.research.ed.ac.uk/portal/files/19835682/is2015_evaluation_dataset_description.txt, 2015, [Online; accessed 10-June-2015].