Learning word vector representations based on acoustic counts

Citation for published version:
Ribeiro, S, Watts, O & Yamagishi, J 2017, Learning word vector representations based on acoustic counts. in Interspeech 2017. DOI: 10.21437/Interspeech.2017-1340

Digital Object Identifier (DOI):
10.21437/Interspeech.2017-1340

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Interspeech 2017

Learning word vector representations based on acoustic counts

M, Sam Ribeiro[1], *Oliver Watts*[1], *Junichi Yamagishi*[12]

[1]Centre for Speech Technology Research, University of Edinburgh, UK
[2]National Institute of Informatics, Tokyo, Japan

m.f.s.ribeiro@sms.ed.ac.uk, owatts@inf.ed.ac.uk, jyamagis@inf.ed.ac.uk

Abstract

This paper presents a simple count-based approach to learning word vector representations by leveraging statistics of co-occurrences between text and speech. This type of representation requires two discrete sequences of units defined across modalities. Two possible methods for the discretization of an acoustic signal are presented, which are then applied to fundamental frequency and energy contours of a transcribed corpus of speech, yielding a sequence of textual objects (e.g. words, syllables) aligned with a sequence of discrete acoustic events. Constructing a matrix recording the co-occurrence of textual objects with acoustic events and reducing its dimensionality with matrix decomposition results in a set of context-independent representations of word types. These are applied to the task of acoustic modelling for speech synthesis; objective and subjective results indicate that these representations are useful for the generation of acoustic parameters in a text-to-speech (TTS) system. In general, we observe that the more discretization approaches, acoustic signals, and levels of linguistic analysis are incorporated into a TTS system via these count-based representations, the better that TTS system performs.

Index Terms: speech synthesis, text-to-speech, vector representations, word embeddings, deep neural networks

1. Introduction

In statistical parametric speech synthesis, acoustic parameters are generated by an acoustic model and then used to drive a vocoder in order to obtain an artificially-generated speech waveform. The acoustic model has, in recent years, typically taken the form of a deep neural network (DNN) [1, 2]. The input to this model is often referred to as the *linguistic specification*, which is a representation designed to bridge the gap between text and speech. Common feature sets for English data, such as the one described in [3], mostly involve context-dependent phones, syllable stress, word part-of-speech, as well as various positional features describing phonetic and prosodic contexts within a text sentence. The group of modules that processes a text sentence and generates the corresponding linguistic specification is often termed the *front-end*.

However, earlier work in the context of HMM-based speech synthesis found that features defined at linguistic levels above the syllable have little impact on the prediction of acoustic parameters [4]. Good representations of higher-level phenomena (often related to syllables or words) are essential for accurate generation of natural speech prosody, especially in the context of expressive audiobook speech synthesis, where speech is expected to be more fluid and pleasing.

In this work, we investigate a method to learn acoustically-motivated representations of words and syllables for text-to-speech synthesis. For this task, we explore vector space models (VSM), which are a well-established approach for obtaining semantic representations in the field of Natural Language Processing (NLP). VSMs are rooted in the distributional hypothesis, which claims that words that have similar contexts tend to have similar meanings [5, 6]. The approaches chosen to learn such representations can be grouped into two main classes. These can be termed, following the terminology of [7], *count models* and *predictive models*.

The first class of models is defined by extracting co-occurence statistics over large text corpora. Various transformations can be applied to the raw counts, such as context weighting or dimensionality reduction techniques [8, 9, 10]. Conversely, the second class of models frames the problem as a context prediction task. That is, given a word, it is the model's objective to determine the context with which it occurs. Therefore, it is expected that words that have similar contexts will be mapped to similar representations in the low-dimensional dense space learned by the model [11, 12, 13, 14]. Investigations have been made into these two approaches, comparing them with various configurations on a set of semantic tasks. Although earlier work showed a clear preference for predictive models [7], recent work showed that their superiority might not be as obvious [15].

In terms of their application to speech synthesis, various approaches have been proposed. Representations learned with count-based methods have been explored as input features to modules within a TTS front-end (e.g. phrase-break prediction [16]), replacement of those modules (e.g. part-of-speech tagging [17]), or as direct input for acoustic modelling [17, 18, 19]. With recent developments in neural network architectures, predictive approaches have gained popularity. Recent work investigated representations of words derived from large text databases [20, 21] and in combination with acoustic parameters [22].

In this work, we investigate a simple approach inspired by the traditional class of models based on co-occurence statistics. Such statistics are extracted over a parallel corpus of text and speech and common transformations are applied to the raw count matrices. In a real-world scenario, these representations could be easily included in the *front-end* of a text-to-speech system as simple look-up tables. Section 2 reviews the methodology proposed for learning count-based word vector representations. Section 3 defines the dataset used, while Section 4 details a set of experiments investigating the effect of the learned representations on a text-to-speech acoustic model. A perceptual evaluation is described in Section 5, followed by a discussion of the results in Section 6.

2. Count-based representations

Given a fixed vocabulary V and a fixed set of acoustic classes A, we define a count matrix $M \in \mathbb{R}^{|V|\times|A|}$. M_{ij} denotes the number of times the j^{th} class is observed occurring with the i^{th} vocabulary unit. The vocabulary can be defined over textual objects (e.g. words, syllables). The classes can be defined by discretizing an acoustic signal, such as *f0* or energy. Sections 2.1 and 2.2 provide details on how these classes are determined.

Because occurrences can be context-dependent, we can extend the set of classes over a unit type to account for neighboring occurrences of the acoustic class. If we set a window of size w, then $M \in \mathbb{R}^{|V| \times w|A|}$. For example, consider an utterance for which U is a sequence of linguistic units and C is the corresponding sequence of acoustic classes. If $w = 3$, then at timestep t we count the occurrence of C_{t-1}, C_t, and C_{t+1} in the i^{th} row of M, for which i is the vocabulary index of the unit U_t. Note that, in this case, the tokens U_{t-1} and the U_{t+1} are not used for the counts of U_t. Each row of the raw count matrix M is then normalized by the total number of counts within each sub-vector of occurrences. Therefore, each sub-vector of the i^{th} token's row is a probability distribution over the acoustic classes A. Each token's row consists of the concatenation of w probability distributions.

Finally, we reduce the dimensionality of the normalized count matrix M by finding the Singular Value Decomposition (SVD) of the matrix, such that: $M = U \Sigma V^T$. We take k left singular vectors of M, such that the sum of squares of the retained singular values is at least 90% of the sum of squares of all singular values. The result of this operation is a matrix $\hat{U} \in \mathbb{R}^{|V| \times k}$. Each row of this matrix corresponds to a entry in the vocabulary V, and we let that be the representation for that linguistic unit.

2.1. Cluster-based class definition

This section describes a possible approach to the quantization of an acoustic signal into a set of classes A. We assume we are given a set of linguistic units, corresponding to entries in a vocabulary V, and its acoustic signal, such as *f0*, with known unit boundaries.

Within each utterance, the signal is normalized to zero mean and unit variance. For each unit, the Discrete Cosine Transform (DCT) is applied to the samples associated with the linguistic unit. The DCT stylizes a signal of N discrete samples with a weighted sum of zero phase cosine functions. The signal is represented by N DCT coefficients: $[c_1, c_2, c_3, ...c_N]$. Most of the energy is stored in the initial coefficients, which often leads to an approximation of the signal with minimal loss by truncating the coefficients to the first d samples.

At this point, each unit is assigned a vector with d components representing its *f0* signal. We then use k-means clustering to group the acoustic vectors into classes. For clustering, we exclude the zeroth DCT coefficient, as that is approximately the mean energy of the signal and can heavily bias the clustering step. We can regard the clustered vector as a representation of the *shape* of the signal for a given linguistic unit. The acoustic classes A are defined to be the clusters identified by k-means. An additional class is added to represent silences such as pauses or hesitations. Figure 1 illustrates the average *shape* for four sample clusters using this method.

2.2. Mean-based class definition

The cluster-based representations ignores the mean value of the unit when defining the acoustic classes. Therefore, a simpler approach quantizes the mean value of the signal over the entire linguistic unit. If we consider the *f0* signal, we might observe that a speaker's range is mostly within 100-300Hz, as shown in Figure 1. We then define 100 classes over this interval, each spanning a range of 2Hz. Two additional classes are added to include occurences below and above the interval. An additional class is added to represent silences. Note that there are several hyperparameters required by the two proposed class definitions, such as number of clusters, number of retained DCT coefficients, and bin size. Details of hyperparameter choices are given in Section 4.

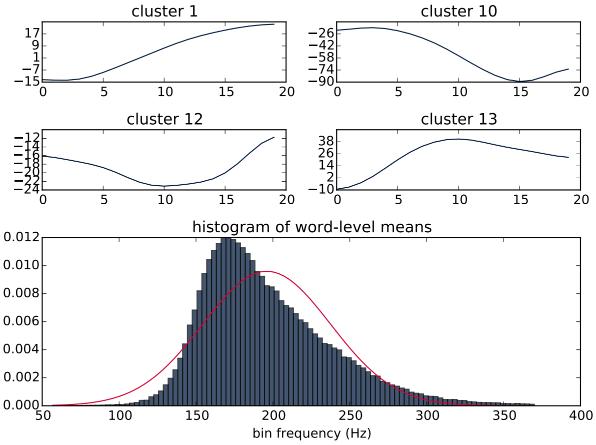

Figure 1: *The top figures illustrate the average DCT vector for four sample clusters, reconstructed with 20 samples. The bottom figure shows a normalized histogram of* f0 *means at word-level with a best fit line.*

3. Data

We use the data made available to the Blizzard Challenge 2013 [23], provided by Lessac Technologies Inc. and originally available from Voice Factory International Inc. The data consists of a single female speaker reading the text of classic novels. This database is of particular interest as the speaker is a professional narrator and actress, which suggests the prosodic variation correlates meaningfully with the text being read. It is also a large dataset, which is interesting for this type of study. However, as the speaker mimics character voices over several books, there is a large variance in terms of speaking styles. Therefore, utterance selection using an active learning approach [24, 25] was performed and a subset of utterances corresponding mostly to narrated speech were selected.

Given the utterance-level segmentation already available from the Blizzard challenge, state-level forced aligment was obtained using context independent HMMs with Festival[1] and HTK[2] via the Merlin toolkit [26]. Pauses were inserted motivated by acoustic evidence, using Festvox's *ehmm*[27]. The training set consists of approximately 18 hours of speech over 13000 utterances, with approximately 220k word and 300k syllable tokens. We set aside an additional 300 and 100 utterances for validation and test purposes.

4. Experiments

4.1. Baseline

The baseline system is a simple feedforward multilayer perceptron. A network of 6 hidden layers, each with 1024 nodes, is used. The hidden layers use *tanh* as the activation function and the output layer uses a linear activation function. For training, mini-batch size is set to 256 and we set a maximum number of epochs to 25 with 5 warmup epochs. Learning rate is initially set to 0.002 for warmup epochs and after that reduced by 50%

[1] `http://www.cstr.ed.ac.uk/projects/festival`

[2] `http://htk.eng.cam.ac.uk`

Table 1: *Objective results for count-based representations at word and syllable levels for fzero and energy signals, counting over classes defined over means or clustered vectors. Dimension indicates the dimensionality of the representation on the decomposed count matrices. Input dimension denotes the dimensionality of the input features to the network, which includes a window of 3 units. MCD is mel-cepstral distortion, BAP is band aperiodicity error, and RMSE and CORR are the root-mean-squared error and correlation between predicted and original* f0 *signals on voiced frames only. Systems marked with an asterisk (*) are used for a listening test.*

Unit	Signal	Classes	Dimension	Input dimension	MCD	BAP	F0-RMSE	F0-CORR
baseline*	-	-	-	594	5.717	2.538	38.137	0.455
word	fzero	clusters	50	744	5.688	2.531	37.629	0.472
word	fzero	mean	150	1044	5.673	2.520	**37.095**	**0.483**
word	fzero	cluster+mean	200	1194	5.656	**2.516**	37.263	0.473
word	energy	clusters	50	744	5.692	2.521	38.217	0.452
word	energy	mean	100	894	5.690	2.529	38.211	0.473
word	energy	cluster+mean	150	1044	5.680	2.527	38.245	0.462
word*	fzero+energy	cluster+mean	350	1644	**5.637**	**2.517**	37.194	**0.479**
syllable	fzero	cluster+mean	180	1134	5.686	2.527	38.018	0.476
syllable	energy	cluster+mean	150	1044	5.673	2.52	38.029	0.474
syllable	fzero+energy	cluster+mean	330	1584	**5.645**	**2.505**	37.264	**0.483**
word+syllable*	fzero+energy	cluster+mean	680	2634	**5.612**	**2.501**	**36.927**	**0.498**

with each epoch. Momentum is set to 0.3 for warmup epochs and 0.9 for all others. L2 regularization weight is set to 10^{-5}. Training is done with the Merlin Toolkit [26].

For output features, we use *log-f0*, 60-dimensional mel cepstral coefficients (MCCs), and 25 band aperiodicities (BAPs) at 5 ms intervals, extracted using STRAIGHT [28]. To these features, we append their respective dynamic features (deltas and delta-deltas). The *log-f0* signal is linearly interpolated through unvoiced regions and a binary voiced/unvoiced decision is appended to the acoustic feature vector. The complete output vector has a total of 259 dimensions, which are then normalized to zero mean and unit variance.

Input features to the network are derived from the labels extracted with Festival and they correspond to a set of 592 binary questions defined at phone, syllable, and word levels. These are quinphone identity, syllable stress, and guessed part-of-speech, as well as all positional features. To these questions, we append 2 features indicating frame number relative to phone size and state number. We let this standard feature set be the input to the baseline system. This is a standard system for TTS using a commonly used feature set. Remaining models are trained under identical conditions, but *append* the learned representations to the baseline feature set, using a window of 3 units. That is, if using word-level features, we append the representations of previous, current, and next word. All input features are normalized to the range [0.01, 0.99]

Additional baselines are not included as earlier work failed to observe improvements when using word vector representations directly with an acoustic model [21, 22]. Some improvements were observed when learned vectors were used to replace knowledge-based features [18, 19]. For the moment, we do not evaluate our method under this scenario.

4.2. Fundamental Frequency

In this set of experiments, we learn representations at word-level using the *f0* signal. The two approaches described in Sections 2.1 and 2.2 are used. The training data contains approximately 220k word tokens. The vocabulary is defined by taking all words types that occur at least 5 times. All other words are mapped to a token *UNK* symbolizing out-of-vocabulary entries. This generates a set of units V with 4468 word types. With this vocabulary, we map 8.7% of the total tokens to *UNK*.

For cluster-based representations, we set the number of DCT coefficients to 8, excluding the zeroth coefficient. We then use k-means to map the DCT coefficients at word-level into 20 clusters. An additional class accounts for silence or pause tokens. This gives us the set of acoustic classes A. For mean-based representations, we set the *f0* range to be between 100Hz and 300Hz. With a bin size of 2Hz, this gives us 100 acoustic classes. To these we append 2 additional classes for any word-means occurring above or below the range and 1 additional class for silence or pause tokens. Figure 1 shows an example of 4 clusters and the histogram of word means for the *f0* signal.

We set the count window w to be 3, which results in a count vector of size $w|A|$ for each word. Note that silence or pause tokens are not in the vocabulary V, but they are taken into account because $w > 1$. Their counts participate in the neighboring acoustic classes of in-vocabulary units. We also consider representations using both cluster and mean-based class definitions. In this case, counts are extracted and normalized separately and concatenated to form the matrix M. We then find the SVD of this matrix to get the reduced matrix $\hat{U}$.

Table 1 details objective results using the *f0* signal and both count methods. We observe improvements with respect to the baseline with all *f0*-based representations. In terms of the two proposed methods, the mean-based approach appears to outperform the cluster-based representations. Although surprisingly their interaction improves MCD, it does not perform as well as the mean-based approach on *f0* RMSE and correlation.

4.3. Energy

The type of representation proposed in this paper can exploit any type of acoustic signal. We experiment with the zeroth mel-cepstral coefficient, which may be regarded as a measure of the energy of a speech frame. Table 1 details objective results for systems appending representations learned with the zeroth mel-cepstral signal. The same details described in Section 4.2 are used for these representations. Cluster-based representations use the same hyperparameters. Mean-based representations use 80 classes over the range $[3, 7]$ with a bin size of 0.05, and we include 3 additional classes. As before, the system combining both count approaches concatenates the normalized counts before applying SVD. For the system combining both signals (*fzero+energy*), we use SVD to produce two separate matrices $\hat{U}_{fzero}$ and $\hat{U}_{energy}$ and we simply concatenate the learned representations to the baseline system's features.

As expected, using the zeroth mel-cepstral signal provides little improvement in terms of *f0*. However, quite surprisingly, it does not outperform representations based on the *f0* in terms of mel-cepstral distortion. No clear difference is observed in terms of the mean and cluster-based methods, but we again observe slight improvements through their interaction. Combining both signals results in the best improvements of all representations defined at word-level.

4.4. Syllable-level representations

We can easily extend this approach to other types of linguistic units, such as the syllable. We represent syllable types textually as the concatenation of the phones present in a given syllable and we build V by mapping all units with fewer than 5 occurrences over the training data to the unknown token *UNK*. From the approximately 300k tokens, a vocabulary of 3447 unit types is defined. This maps 1.9% of the total tokens to *UNK*. The remaining parameters are similar to those of the word-level representations, except we vary the number of singular vectors kept after SVD. For brevity, we do not include all system combinations and we evaluate only representations using both cluster-based and mean-based approaches. In terms of combination of different counts, discretization methods were concatenated before matrix decomposition. All other levels of variation assume separate matrices, which were then added to the linguistic specification using a window of 3 textual units.

Although we observe some improvements with syllable-level representations, they do not outperform their equivalent system at word-level. As before, the interaction of both *f0* and energy representations shows the best results. The system including representations at both syllable and word levels gives the best results of all configurations. Although there might be some correlation between representations, their interaction is still useful to the acoustic model.

5. Subjective evaluation

Given the large number of system configurations, we opt to conduct a perceptual evaluation on selected systems, based on the results shown in Table 1. Besides the baseline, we consider a system using the best combination of word-level features and a system using the best combination of word and syllable level features. The systems are marked with an asterisk on Table 1. A preference test with a *no preference* option was conducted on the three selected systems. From the test set, 50 utterances were randomly selected and synthesized with the acoustic parameters generated from each system. 20 native speakers judged randomized utterance pairs for each pair of systems. Each utterance pair was judged 10 times and each condition received a total of 500 judgments. Percentage preferences are shown in Table 2, which includes the results of a 1 tailed-binomial test assuming an expected 50% split, with the *no preference* judgments distributed equally over the other two conditions.

The results are consistent with those in Table 1. Systems using the proposed additional features are preferred over the system using no features. Considering multiple linguistic levels (e.g. words, syllables) is preferred over using only representations learned at word level.

Table 2: *Preference test results*

baseline	word	word+syllable	N/P	Binomial test p
38.6%	46.8%	–	14.6%	$p < .05$
36.4%	–	47.0%	16.6%	$p < .05$
–	34.8%	43.0%	22.2%	$p < .05$

6. Discussion and conclusion

To produce the system configurations shown in Table 1, we varied three main factors pertaining to how the representations were learned: *discretization method* (e.g. cluster, mean), *acoustic signal* (e.g. *f0*, energy), and *linguistic level* (e.g. words, syllables). In terms of the first factor, we observe that the mean-based approach outperforms the cluster-based approach, but combining both methods provides better results than either method separately. This pattern is also observed when varying the acoustic signal: *f0* is shown to have a stronger impact on the objective results than energy, but the interaction of both signals gives the best results. In terms of linguistic units, we observe that word representations outperform syllable representations, but combining both levels appears more useful than either method in isolation. This was further validated with a listening tests, as shown in Table 2. In general, we observe that the more information is incorporated into the *linguistic specification* of a TTS system using the proposed representations, the better that TTS system performs.

With respect to the current methodology, it should be noted that no optimization of hyperparameters was attempted. No tuning was performed, for example, on the number of clusters or the number of bins for discretization methods. It was surprising to observe such improvements with fairly arbitrary initial choices of hyperparameters. Further improvements might be observed with careful optimization.

These results are encouraging and suggest further lines of research for this method. In terms of discretization methods, we might consider earlier work for the *f0* signal, such as Tilt [29], MoMel [30], ProsoGram [31], or SLAM [32]. Acoustic signals such as jitter and shimmer might also be useful in the context of a TTS system. Learning representations at multiple levels might be useful, as recent work as shown that the phrase level is particularly relevant [21]. Other lines of research could focus on whether these representations would be useful across speakers. Earlier work failed to observe statistically significant differences with listening tests when appending word vectors directly to the input features of an acoustic model [21, 22] and future work could focus on a deeper analysis of these methodologies, evaluating them as additional features or as replacement of knowledge-based features, such as POS tags.

We have proposed a novel method to learn word and syllable vector representations by taking counts over discretized acoustic events. In general, we have observed that the more discretization approaches, acoustic signals, and levels of linguistic analysis are incorporated into a TTS system via these count-based representations, the better that TTS system performs.

Acknowledgements: This research was supported by the Swiss NSF under grant CRSII2 141903: Spoken Interaction with Interpretation in Switzerland (SIWIS), the EPSRC Standard Research Studentship (EP/K503034/1), and the EPSRC Standard Research Grant EP/P011586/1: *Speech Synthesis for Spoken Content Production (SCRIPT)*.

7. References

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

[2] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, "On the training aspects of deep neural networks (dnn) for parametric tts synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014.

[3] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[4] M. Cernak, P. Motlicek, and P. N. Garner, "On the (un) importance of the contextual factors in HMM-based speech synthesis and coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8140–8143.

[5] A. Lenci, "Distributional semantics in linguistic and cognitive research," *Italian journal of linguistics*, vol. 20, no. 1, pp. 1–31, 2008.

[6] P. D. Turney, P. Pantel *et al.*, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, no. 1, pp. 141–188, 2010.

[7] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors." in *ACL (1)*, 2014, pp. 238–247.

[8] J. A. Bullinaria and J. P. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behavior research methods*, vol. 39, no. 3, pp. 510–526, 2007.

[9] ——, "Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd," *Behavior research methods*, vol. 44, no. 3, pp. 890–907, 2012.

[10] R. Lebret and R. Collobert, "Rehabilitation of count-based models for word vector representations," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2015, pp. 417–429.

[11] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 384–394.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[13] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 873–882.

[14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, 2014.

[15] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, 2015.

[16] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 2599–2603.

[17] O. Watts, J. Yamagishi, and S. King, "Unsupervised continuous-valued word features for phrase-break prediction without a part-of-speech tagger." in *INTERSPEECH*, 2011, pp. 2157–2160.

[18] O. Watts, "Unsupervised learning for text-to-speech synthesis," Ph.D. dissertation, The University of Edinburgh, 2012.

[19] H. Lu, S. King, and O. Watts, "Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis," *Proc. ISCA SSW8*, pp. 281–285, 2013.

[20] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Word embedding for recurrent neural network based TTS synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2015. ICASSP 2015.*, 2015.

[21] X. Wang, S. Takaki, and J. Yamagishi, "Investigation of using continuous representation of various linguistic units in neural network based text-to-speech synthesis," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 10, pp. 2471–2480, 2016.

[22] ——, "Enhance the word vector with prosodic information for the recurrent neural network based tts system," in *Proc. Interspeech*, San Francisco, United States, September 2016.

[23] S. King and V. Karaiskos, "The blizzard challenge 2013," in *Proc. Blizzard Challenge Workshop*, 2013.

[24] L. C. Yong, O. Watts, and S. King, "Combining lightly-supervised learning and user feedback to construct andimprove a statistical parametric speech synthesizer for malay," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 11, no. 11, pp. 1227–1232, 2015.

[25] O. Watts, A. Stan, R. A. 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 *SSW*, 2013, pp. 101–106.

[26] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," *Proc. SSW, Sunnyvale, USA*, 2016.

[27] K. Prahallad, A. W. Black, and R. Mosur, "Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 2006, pp. I–I.

[28] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.

[29] P. Taylor, "The tilt intonation model," in *International Conference on Spoken Language Processing*, 1998, pp. 1383–1386.

[30] D. Hirst, A. Di Cristo, and R. Espesser, "Levels of representation and levels of analysis for the description of intonation systems," in *Prosody: Theory and experiment*. Springer, 2000, pp. 51–87.

[31] P. Mertens, "The prosogram: Semi-automatic transcription of prosody based on a tonal perception model," in *Speech Prosody 2004, International Conference*, 2004.

[32] N. Obin, J. Beliao, C. Veaux, and A. Lacheret, "Slam: Automatic stylization and labelling of speech melody," in *Speech Prosody*, 2014, pp. 246–250.