INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Enhance the word vector with prosodic information for the recurrent neural network based TTS system

Xin Wang[1,2], *Shinji Takaki*[1], *Junichi Yamagishi*[1,2,3]

[1]National Institute of Informatics, Japan
[2]SOKENDAI University, Japan
[3]University of Edinburgh, UK

wangxin@nii.ac.jp, takaki@nii.ac.jp, jyamagis@nii.ac.jp

Abstract

Word embedding, which is a dense and low-dimensional vector representation of word, is recently used to replace of the conventional prosodic context as an input feature to the acoustic model of a TTS system. However, these word vectors trained from text data may encode insufficient information related to speech. This paper presents a post-filtering approach to enhance the raw word vectors with prosodic information for the TTS task. Based on a publicly available speech corpus with manual prosodic annotation, a post-filter can be trained to transform the raw word vectors. Experiment shows that using the enhanced word vectors as an input to the neural network-based acoustic model improves the accuracy of the predicted F0 trajectory. Besides, we also show that the enhanced vectors provide better initial values than the raw vectors for error back-propagation of the network, which results in further improvement.

Index Terms: Text-to-speech, word embeddings, neural network, prosodic labeling

1. Introduction

A text-to-speech (TTS) system converts text strings into speech waveforms. In order to model the non-linear relationships between text and speech, a TTS system can generally be decomposed into the front- and the back-end. The front-end derives linguistic representations that contain the pronunciation of individual words and the prosody of the input text. Based on the intermediate representations, the back-end acoustic model predicts acoustic features and synthesizes speech waveforms.

In the back-end, the hidden Markov model (HMM) is the classical model for acoustic modeling. However, its limitations such as data fragmentation caused by decision-trees [1] and limited capabilities of Gaussian distributions in HMM states [2] have motivated researchers to use neural networks (NNs) to either complement [2] or replace the HMM-based framework[1, 3, 4]. In the front-end, modules based on decision-trees or other relatively simple models are widely used for grapheme-to-phoneme conversion, part-of-speech (POS) tagging, and symbolic prosodic label prediction [5]. Leveraging NN for these sub-modules may further improve TTS systems [6].

NN models usually require a large amount of manually annotated data for supervised training. Because data annotation such as the prosodic annotation on a speech corpus can be time consuming, combining unsupervised or semi-supervised training schemes with NN models may be more practical. For example, Wang et al. took an unsupervised approach and replaced the conventional prosodic context in TTS systems with vector representations of words learned by NN language models [7].

Although word vectors are shown to be effective in various natural language processing tasks [8], implicit linguistic regularity encoded in these vectors may still be insufficient and noisy for the TTS task. Typically, our previous experiments implied that, at least on the utilized speech corpus, word vectors were not significantly better than the automatically derived prosodic symbols for TTS systems with a acoustic model based on either the recurrent neural network (RNN) or the deep feed-forward neural network (DNN) [9].

Thus, as presented in this paper, we investigated a new semi-supervised approach to find whether task-specific information could improve word vectors for TTS. This approach used a post-filter to transform the raw word vectors into enhanced ones that were expected to encode more 'prosodic' information. The post-filter was implemented as a neural network and trained on a small publicly available corpus with manually annotated prosodic tags. The training scheme, similar to the joint-embedding framework [10], aimed at learning the non-linear relationship between the raw word vector and a prosodic feature vector. This prosodic feature vector was extracted from the hidden layer of another prosodic labeling model that predicted the prosodic tag of a word according to its acoustic features. The experimental results revealed that, enhanced vectors as the input the NN-based acoustic model increased the accuracy of the predicted F0 trajectories. Besides, by updating enhanced vectors as part of the acoustic model through error back-propagation on the large speech synthesis corpus without prosodic annotation, the enhanced vectors could further be improved. However, the improvement on objective measure didn't lead to significant difference in perception tests.

In the rest of this paper, section 2 briefly introduces existing work using the (raw) word vectors for TTS systems. Section 3 explains the proposed approaches to enhance word vectors, including the prosodic labeling model to extract prosodic feature vectors, the post-filter that tunes the word vectors, and the use of enhanced vectors in TTS systems. Section 4 shows the experiments and results, and Section 5 summarizes this work.

2. TTS systems using raw word vectors

Word vectors are dense and low-dimensional continuous representations of words. They can be derived from plain text without task-specific annotation while encode syntactic and semantic regularities of language [12]. It has been shown that word vectors can be plugged into natural language processing (NLP) systems and improve their performance in various tasks including sentence chunking and name entity recognition [8].

Similarly to the NLP approach, word vectors have been

http://dx.doi.org/10.21437/Interspeech.2016-390

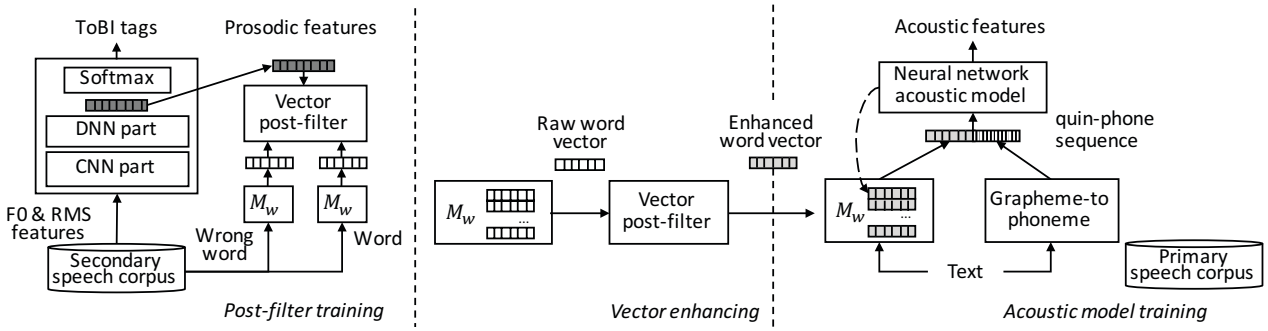

Figure 1: Steps to build the TTS system with enhanced word vectors. A post-filter is trained and used to enhance raw vectors. Enhanced vectors are then used in acoustic modeling. These vectors can be further updated in acoustic modeling (indicated by the dashed arrow). $\boldsymbol{M}_w \in R^{D \times |V|}$ is the vector matrix, where $|V|$ is vocabulary size of $\boldsymbol{M}_w$ and D is vector's dimension. The secondary speech corpus contains manually annotated Tones and Break Indices (ToBI) tags while the large primary corpus for acoustic modeling does not [11].

used in the TTS system to replace the the conventional prosodic context [7]. The motivation is to reduce manual-annotation costs in building prosodic models for TTS systems in a new language or domain. With word vectors as the input to the acoustic model based on a bi-directional recurrent neural network with long-short term memory units (for short, DBLSTM-RNN [13]), this TTS system was reported to outperform other systems without POS tags nor prosodic context as input information [7].

We have also investigated the effectiveness of word vectors in TTS systems with acoustic models based on DBLSTM-RNN and DNN [9]. Further, we used vector representations of phonemes, syllables and phrases as the input to the NN-based acoustic models. These vectors can be learned in the same manner as word vectors can be learned from plain text using the RNN [14] and other log linear models [15]. However, the subjective evaluation revealed insignificant differences in DBLSTM-RNN-based systems with different types of input vectors. Only phrase vectors achieved significant improvements when they were used together with the conventional prosodic context as the inputs to the acoustic model based on DNN.

3. Proposed method

3.1. Motivation

Although word vectors can encode linguistic regularities, the encoded information is somehow limited. For example, while word vectors performed well in predicting the taxonomic properties of words (e.g., an apple is a fruit), their performance in predicting other properties was much worse [16]. One reason for this is that the learning algorithms of word vectors assume similar vectors to be assigned to words that share similar neighboring words. This linear-context assumption is effective to derive the *association* or *topical similarity* between words, but not other semantic relationships [17].

To enrich the word vector with other kinds of semantic knowledge, the distance between words in a semantic lexicon can be integrated into the optimization function for learning new word vectors [18, 19, 20] or refining the raw word vectors [21]. These approaches increase the similarity between words of similar word types. Another method is to utilize the original algorithm to learn word vectors based on syntactic context [22], which results in word vectors with better syntactic regularities. All the previous work has indicated that it is beneficial to enhance the word vectors with task-related information.

3.2. Word vectors enhanced with prosodic information

For the TTS task, we present the post-filtering approach to enhance the word vector with prosodic information, which is shown in Figure 1. In the post-filter training stage, a prosodic feature vector is extract from a prosodic labeling model for each word in the speech corpus with annotated prosodic tags. Then, the post-filter model is trained given the prosodic feature vector and the corresponding raw word vector. This trained post-filter model can be used to enhance any input raw word vectors. After that, the enhanced vectors can be plugged into the acoustic model of a TTS system.

This method is semi-supervised because the speech corpus used in the post-filter training stage contains annotated prosodic tags. However, this corpus is small and different from the main large speech synthesis corpus to train the acoustic model. Typically, the main large corpus contains no prosodic annotation. For brevity, we call the small corpus the secondary corpus and the larger one the primary corpus [11]. The secondary corpus can be a subset of the primary corpus if manual prosodic annotation can be conducted for this subset. Or, in our case, it can be another corpus released with prosodic annotation. The risk of the latter approach is that the prosodic patterns of the two corpora may be different.

3.2.1. Prosodic feature extraction

Among possible ways to extract the prosodic features, we present one approach with an auxiliary task for prosodic labeling. Prosodic labeling aims at predicting symbolic prosodic tags based on the acoustic features of a linguistic unit (e.g., a word or syllable). Research on this topic suggests that statistical prosodic labeling models can learn meaningful relationships between prosodic tags and acoustic features [23, 24]. Thus, we assume that useful features can be extracted by these models.

The utilized prosodic labeling model predicts prosodic tags at the word level. The output targets are the pitch accents defined in Tones and Break Indices (ToBI), a prosody annotation protocol for American English [25, 26]. However, the full set of pitch accents is merged into 5 categories due to the limited size of the secondary speech corpus: the H*, !H* and L* form the first three categories, bitonal accents the forth category, and other symbols the fifth category [27]. The input to the prosodic labeling model is a set of acoustic features extracted from the waveform of a word, including the continuous wavelet transforms of the F0 trajectory [28] and the root mean square (RMS)

level of each speech frame.

The neural network in the prosodic labeling model includes a convolutional neural network (CNN) that extracts a compact representation from the input acoustic features and a feed-forward network that transforms the compact representation into the target prosodic tags. Because the neural network is expected to extract structural features from the input data, we assume that the vector exported by the last hidden layer encodes abstract acoustic information optimized for the prosodic labeling task. This prosodic feature vector can be extracted for each word in the secondary speech corpus after the prosodic labeling model is trained on this corpus. Details on the configuration and hyper-parameters will be provided in Section 4.1.

3.2.2. Post-filter training

The post-filter part is based on a feed-forward neural network. The target feature is the extracted prosodic feature vector from the prosodic labeling model while the input is the raw word vector. To train the model, the triplet ranking loss [10] below is utilized as an objective function E:

$$E = \max\left[0, 1 - Sim(\boldsymbol{p}_w, \mathcal{F}(\boldsymbol{m}_w)) + Sim(\boldsymbol{p}_w, \mathcal{F}(\boldsymbol{m}_{w^-}))\right]. \quad (1)$$

Here, $\mathcal{F}$ is the post-filter model, $\boldsymbol{m}_w$ is the raw vector of word w, $\boldsymbol{p}_w$ is the prosodic feature of w given by the prosodic labeling model, and $sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{||\boldsymbol{x}||\cdot||\boldsymbol{y}||}$ is the similarity between vectors. This criterion also involves the word vector $\boldsymbol{m}_{w^-}$ of a wrong word w^-. This wrong word is currently randomly sampled from the corpus.

The way to train the post-filter model is similar to the joint embedding approach [10]. There, the triplet ranking loss is used to train a NN model that enhances the letter-based word representation with the acoustic information. The main difference is that the acoustic information in their case is the segmental information associated with the word. In our case, we assume the suprasegmental information would be encoded.

3.3. Usage of enhanced word vectors in TTS systems

The enhanced word vectors can be used as input features to the acoustic model in a TTS system, as shown in Figure 1. Because the prosody of a word and its acoustic realization depend on the context beyond the current word [29], the DBLSTM-RNN-based acoustic model is used so that the dependency between the input vectors over a large segment can be learned.

The word vector $\boldsymbol{m}_{w_n}$ of a word w_n can be written as $\boldsymbol{m}_{w_n} = \boldsymbol{M}_w \boldsymbol{v}_{w_n}$, where $\boldsymbol{v}_{w_n}$ is the one-hot representation of w_n. In this sense, $\boldsymbol{M}_w$ can be regarded as the parameter of one layer of the NN-based acoustic model. Thus, $\boldsymbol{M}_w$ can be further updated during the acoustic model training stage by error back-propagation. Although a similar strategy can be used to update the raw word vectors [7], the difference is the initial condition. Due to the difficulties in optimizing the parameters of the lower layer of a deep neural work [30], the word vectors enhanced with prosodic features may be better than the raw vectors as the initial value for back-propagation.

4. Experiments

4.1. Prosodic feature extraction and post-filter training

The Boston University Radio News Corpus (BURNC) [31] was used as the secondary corpus to train the post-filter. The speech data of speaker *f2b* was used because of the manual ToBI annotation provided. This data set contained 148 utterances with 11211 words, among which 6074 words were annotated with pitch accents.

The frame width for acoustic feature extraction was 20 ms and the frame shift was 5 ms. The F0 trajectory was extracted by merging the outputs from multiple F0 extractors; then, it was transformed into a continuous wavelet representation with five sub-components [28]. The root mean square (RMS) of the waveform was calculated using the SPTK toolkit [32], and then normalized [24]. Given the time alignment information in BURNC, the acoustic feature matrix of a word was composed based on the wavelet representation of F0 and the RMS per frame. Finally, the zero padding strategy was used to ensure that each word had 160 frames (only 1.7% of words are longer than 160 frames). Thus, the size of the acoustic feature matrix for one word was 160×6. The acoustic matrices of the central word and its two neighbors were fed as inputs into the prosodic labeling model.

The CNN layer at the bottom of the prosodic labeling model contained 10 feature filters with a receptive field size of 5×6; the max pooling stride was $(10, 1)$. This configuration ensured that the five F0 wavelet components and the RMS trajectory were separately transformed by the CNN layer. Feed-forward layers with the size of $(500, 320, 80)$ were added after the CNN layer. The output layer was the softmax layer. This network structure was selected based on experiments on prosodic labeling tasks. If we change the targets and train the model to predict the presence/absence of accent and boundary tone, we can get the results comparable to existing work [23, 24] as Table 1 shows. Thus, this network can be expected to extract useful feature for our task. The size of the last hidden layer was 80, which ensured that the dimension of the extracted prosodic vector was compatible with the word vectors to be enhanced.

After training the prosodic labeling model, we extracted the feature vectors for all words in *f2b*. These vectors were used as the target of the vector post-filter. Then, we used the same vector set in Wang et al. [7] as the input raw word vectors. This vector set contained 80-dim vectors for 82390 words. We additionally calculated the average of the word vectors to represent unseen words. Through experiments on *f2b* data, we selected the network with 2 hidden layers of size $(160, 160)$ as the vector post-filter. The input and output dimension was 80. The post-filter was trained by stochastic gradient descent with respect to the triplet-ranking loss for 1000 epochs, which increased the average similarity score between the input and output vectors from 0.091 to 0.467. Both the prosodic labeling model and the post-filter were implemented using the Theano library [33].

Note that, out of the 82390 word types covered by the word vector set, only 2792 of them existed in the *f2b* data. We assumed that the post-filter learned from the *f2b* data could be generalized to other words in the word vector set. A related idea dealing with "out-of-vocabulary (OOV)" words has been presented by Tafforeau et al. [34]. Different from their approach based on linear interpolation, we utilized the non-linear regression provided by the neural network to derive the representation for OOV words.

Table 1: Performance of the prosodic labeling model in speaker-dependent (*f2b*) task that predicts presence/absence of accent and intonational phrase boundary (IPB) on word-level.

	Precision	Recall	f_1-score	Accuracy
Accent	0.901	0.863	0.882	0.869
IPB	0.701	0.805	0.749	0.895

Table 2: Lists of experimental systems

ID	Input to the acoustic model besides the quin-phone
R_N	No prosodic context or word vectors
R_p	Prosodic context given by Flite
R_{wr}	Raw word vectors
R_{we}	Enhanced word vectors
$R_{wr_{bp}}$	Raw word vectors after back-propagation in R_{wr}
$R_{we_{bp}}$	Enhanced vectors after back-propagation in R_{we}

4.2. Experiments on TTS task

We carried out experiments on the TTS task based on enhanced word vectors. Here, we use we and wr to denote the enhanced and raw word vectors. The database for the acoustic model training contained 12072 English utterances (16 hours) by a female speaker in a neutral news reading style. Both the test and validation set contained 500 randomly selected utterances. Mel-generalized cepstral coefficients (MGC) of order 60, a one-dimensional continuous F0 trajectory, the voiced/unvoiced (V/U) condition, and band aperiodicity of order 25 were extracted for each speech frame by the STRAIGHT vocoder [35]. The F0 trajectory was further converted to Mel-scale according to $m = 1127 * \log(1 + f/700)$. Delta and delta-delta components of the acoustic features were used for all the systems. The Flite toolkit [36] conducted the grapheme-to-phoneme conversion for both the training and test sets. The phonemic information given by the Flite only contained the phoneme identity (quin-phone), without other numerical information such as the position of the current phoneme.

The experimental systems are listed in Table 2. All the experimental systems only took F0 as the output feature. They all adopted the neural network with two feed-forward and two D-BLSTM layers with the layer size as (512,512,256,64). Another DBLSTM-based system was trained to predict spectral features from all the experimental systems. This experiment setup was motivated by our observation that word vectors can not increase the accuracy of predicted spectral features significantly on the same corpus [9]. Another concern is that, with spectral and F0 as the target features, the neural network may devote most of its capability to modelling the spectral features [37]. Thus, with F0 as the sole output feature, the usefulness of enhanced word vectors can be reflected better. All the acoustic models were implemented based on a revised CURRENNT toolkit [38] [1].

Objective measures included the root mean square error (RMSE) and correlation against the natural F0 trajectory. The duration were copied from the natural data (at the phoneme level). The results on the test set are presented in Figure 2. As R_{wr} and R_{we} achieve better objective measure than R_N, this indicates word vectors contain useful information for F0 modelling. Moreover, R_{we} is better than R_{wr}, which suggests that enhanced word vectors contain more task-related information. $R_{wr_{bp}}$ and $R_{we_{bp}}$ can further increase the accuracy of prediction by updating word vectors through back-propagation. However, $R_{we_{bp}}$ results in higher accuracy than $R_{wr_{bp}}$, possibly due to the better initial condition provided by enhanced vectors. All results indicate that enhanced vectors are more suitable for the TTS task. Note that R_p is worse than R_{we} and R_{wr} due to the noisy prosodic context automatically generated by Flite.

Subjective A/B preference tests with 20 native English speakers were conducted for the three pairs of systems shown in Figure 3, where each test contained 40 pairs of synthetic samples. Although the difference is trivial, the system $R_{we_{bp}}$ is judged to be better that R_p. Comparisons between R_{wr} and R_{we} and between R_{we} and $R_{we_{bp}}$ indicate that the main contribution to the performance of $R_{we_{bp}}$ comes from the prosodic information encoded in the vectors. However, further updating the enhanced vectors based on back-propagation may not lead to further perceptible improvement on synthetic speech. Note that the difference between R_N and R_p is not significantly different based on our previous work [9].

[1] Toolkit and speech samples available on http://tonywangx.github.io

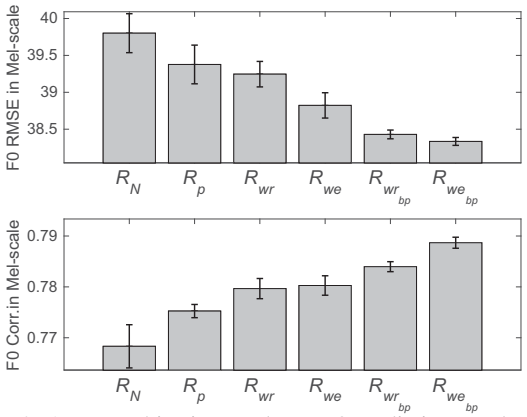

Figure 2: Average objective results on F0 prediction. Each system was trained twice with random initialization, and the models given by the last five training epochs of each trial were used to predict F0. The error bar shows the standard deviation of the ten sets of results for each system.

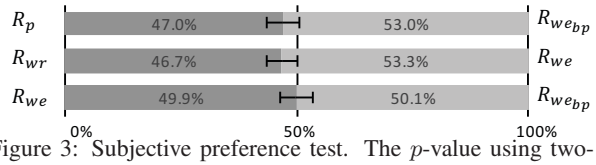

Figure 3: Subjective preference test. The p-value using two-tailed binomial test are 0.097, 0.071, 0.972. The p-value using one-sample t-test are 0.089, 0.066, 0.944.

5. Conclusion

This paper presented one way of enhancing the word vectors through pushing the raw word vectors towards the prosodic feature vectors extracted by a CNN-based prosodic labeling model. The enhanced word vectors can be directly fed into the acoustic model or further tuned in the acoustic model training stage. The experiments demonstrated that the enhanced word vector increased the accuracy of the predicted F0 trajectory. However, the improvement does not lead to significant difference in subjective evaluation test.

As one reviewer points out, the prosodic patterns can be speaker dependent. One future work is to annotate part of the primary corpus with prosodic tags and then test the proposed method. Another possible work is to train the post-filter in an speaker-independent way.

6. Acknowledgements

We thank the reviewers for the critical comments. This work was partially supported by EPSRC through Programme Grant EP/I031022/1 (NST) and EP/J002526/1 (CAF) and by the Core Research for Evolutional Science and Technology (CREST) from the Japan Science and Technology Agency (JST) (uDialogue project). Shinji Takaki was supported in part by the NAVER Labs..

7. References

[1] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *ICASSP-2013*, 2013, pp. 7962–7966.

[2] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 2129–2139, 2013.

[3] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," *INTERSPEECH-2014*, pp. 1964–1968, 2014.

[4] S. Kang and H. M. Meng, "Statistical parametric speech synthesis using weighted multi-distribution deep belief network," in *INTERSPEECH-2014*, 2014, pp. 1959–1963.

[5] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[6] C. Ding, L. Xie, J. Yan, and W. Zhang, "Automatic prosody prediction for chinese speech synthesis using BLSTM-RNN and embedding features," in *ASRU-2015*, 2015.

[7] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Word embedding for recurrent neural network based TTS synthesis," in *ICASSP-2015*, 2015, pp. 4879–4883.

[8] 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*, 2010, pp. 384–394.

[9] 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*, (Under review).

[10] S. Bengio and G. Heigold, "Word embeddings for speech recognition." in *INTERSPEECH-2014*, 2014, pp. 1053–1057.

[11] O. S. Watts, "Unsupervised learning for text-to-speech synthesis," Ph.D. dissertation, University of Edinburgh, 2013.

[12] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, 2013, pp. 746–751.

[13] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Technische Universität München, 2008.

[14] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH-2010*, vol. 2, 2010, p. 3.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS-2013*, 2013, pp. 3111–3119.

[16] D. Rubinstein, E. Levi, R. Schwartz, and A. Rappoport, "How well do distributional models capture different types of semantic knowledge?" in *ACL-IJCNLP*, 2015, pp. 726–730.

[17] F. Hill, R. Reichart, and A. Korhonen, "Simlex-999: Evaluating semantic models with (genuine) similarity estimation," *Computational Linguistics*, 2015.

[18] D. Fried and K. Duh, "Incorporating both distributional and relational semantics in word representations," *arXiv preprint arXiv:1412.4369*, 2014.

[19] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T.-Y. Liu, "RC-Net: a general framework for incorporating knowledge into word representations," in *CIKM-2014*, 2014, pp. 1219–1228.

[20] J. Bian, B. Gao, and T.-Y. Liu, "Knowledge-powered deep learning for word embedding," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 132–148.

[21] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," *arXiv preprint arXiv:1411.4166*, 2014.

[22] O. Levy and Y. Goldberg, "Dependency-based word embeddings." in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 302–308.

[23] A. Rosenberg, "Automatic detection and classification of prosodic events," Ph.D. dissertation, Columbia University, 2009.

[24] K. Chen, M. Hasegawa-Johnson, and A. Cohen, "An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model," in *ICASSP-2004*, vol. 1, 2004, pp. I–509.

[25] K. E. A. Silverman, M. E. Beckman, J. F. Pitrelli, M. Ostendorf, C. W. Wightman, P. Price, J. B. Pierrehumbert, and J. Hirschberg, "ToBI: a standard for labeling English prosody," in *ICSLP-1992*, 1992, pp. 867–870.

[26] M. E. Beckman and G. Ayers, "Guidelines for ToBI labelling," *The OSU Research Foundation*, vol. 3, 1997.

[27] A. Black and A. Hunt, "Generating F0 contours from ToBI labels using linear regression," in *ICSLP-1996*, vol. 3, 1996, pp. 1385–1388.

[28] A. S. Suni, D. Aalto, T. Raitio, P. Alku, M. Vainio *et al.*, "Wavelets for intonation modeling in HMM speech synthesis," in *8th ISCA Workshop on Speech Synthesis, Proceedings, Barcelona, August 31-September 2, 2013*, 2013.

[29] J. Cole, "Prosody in context: a review," *Language, Cognition and Neuroscience*, vol. 30, no. 1-2, pp. 1–31, 2015.

[30] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[31] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel, "The Boston University radio news corpus," *Linguistic Data Consortium*, 1995.

[32] SPTK Working Group, "Speech Signal Processing Toolkit (SPTK) Version 3.9," 2015. [Online]. Available: http://sp-tk.sourceforge.net

[33] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[34] J. Tafforeau, T. Artieres, B. Favre, and F. Bechet, "Adapting lexical representation and OOV handling from written to spoken language with word embedding," in *INTERSPEECH-2015*, 2015.

[35] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[36] HTS Working Group, "The English TTS System "Flite+hts_engine"," 2014. [Online]. Available: http://hts-engine.sourceforge.net/

[37] X. Wang, S. Takaki, and J. Yamagishi, "Investigating very deep highway networks for parametric speech synthesis," in *Submitted to SSW-9*.

[38] F. Weninger, J. Bergmann, and B. Schuller, "Introducing CURRENT: The Munich open-source CUDA recurrent neural network toolkit," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 547–551, 2015.