USING CONTINUOUS LEXICAL EMBEDDINGS TO IMPROVE SYMBOLIC-PROSODY PREDICTION IN A TEXT-TO-SPEECH FRONT-END

Asaf Rendel[1], *Raul Fernandez*[2], *Ron Hoory*[1], *Bhuvana Ramabhadran*[2]

[1]IBM Haifa Research Lab, Haifa – Israel
[2]IBM TJ Watson Research Center, Yorktown Heights, NY – USA

asafren@il.ibm.com, fernanra@us.ibm.com

ABSTRACT

The prediction of symbolic prosodic categories from text is an important, but challenging, natural-language processing task given the various ways in which an input can be realized, and the fact that knowledge about what features determine this realization is incomplete or inaccessible to the model. In this work, we look at augmenting baseline features with lexical representations that are derived from text, providing continuous embeddings of the lexicon in a lower-dimensional space. Although learned in an unsupervised fashion, such features capture semantic and syntactic properties that make them amenable for prosody prediction. We deploy various embedding models on prominence- and phrase-break prediction tasks, showing substantial gains, particularly for prominence prediction.

Index Terms— word embeddings, prominence prediction, prosodic phrasing, speech synthesis, deep learning

1. INTRODUCTION

The generation of natural and expressive prosody from text is one of the fundamental tasks in a text-to-speech (TTS) synthesis system. In particular, prosodic phrasing, the proper assignment of prominence, and its suitable acoustic realization, are some of the main challenges in generating artificial speech that is perceived as natural and expressive to the task at hand. One common way in which different synthesis architectures approach this problem is by decomposing it into two stages: First, a linguistic front-end module assigns symbolic intonational phrase-break and prominence labels from purely textual features of the input. This intermediate symbolic prosodic representation is then passed to a back-end to exploit in a variety of ways (for example, as an additional feature when generating an f_0 contour, or as an additional target in a unit-selection module).

In this work, we focus on the first stage of this approach: assigning word-level phrase breaks and prominence from text. This is a task that remains challenging for several reasons. First, the problem is under-determined since a given input text string can result in various symbolic prosodic realizations. Secondly, the linguistic features accounting for such nuanced differences may not be well-understood, be difficult to extract from text (e.g., attitudes), or reside outside the text altogether (e.g., world knowledge). One approach to improve the prediction of symbolic prosody is to refine the linguistic analysis to extract richer, lower-level syntactical and semantic features and discover how they correlate with prosodic realizations. A complementary research inquiry, and the one we follow in this work, is to focus on lexical representations where word identities, or rather representations derivable from them, can serve as features. The naïve approach of using the raw lexical identity directly, however, suffers from serious drawbacks since the high dimensionality of the input vocabulary will lead to data sparsity, particularly in the supervised-training framework we adopt later in the paper when training phrase-break and prominence classifiers. Instead, we seek lexical representations that embed the input words in a (continuous) lower-dimensional manifold.

Continuous word embeddings have recently received a lot of attention in the literature, having been successfully applied to various natural language processing tasks such as parsing, detecting word-analogy and word-similarity relations, and named-entity recognition. In this work we explore their novel use for the prediction of symbolic prosodic labels: prominence and phrase breaks. The models we consider have been trained on large amounts of text data requiring no supervised learning, and have been shown to lead to embeddings with desirable properties, such as close proximity (using Euclidean or cosine distances) for words that are semantically related. The emergence of semantically meaningful clusters derived from large corpora is of particular interest for prosodic modeling, particularly prominence, for the following reason: A prominence-assignment model is likely to be trained with only limited amounts of supervised data that bear prosodic-prominence annotations, and thus have limited ability to discover equivalency classes (for the purpose of assigning prominence) between different words (even if it had access to lexical features). We hypothesize that semantically-related classes may be treated similarly when assigning prosody. Therefore, providing embedding features that capture synonymy relations as inputs during the training stage allows the model to exploit correlations between points in the embedded space and likelihood of prominence. This, in turn, allows words not seen in the prominence-training corpus to be treated similarly to other semantically-related words (which do appear in the vocabulary of the word embedding), so the prominence predictor may show better generalization, particularly when dealing with large or open classes (e.g., names) which are poorly represented in the labeled prosodic corpus.

2. PREVIOUS AND RELATED WORK

There is a substantial literature on automatic classification of prosodic categories going back now by a few decades. The overwhelming focus of this body of work, however, has been on the annotation of speech corpora, where systems have access to both lexical and acoustic features, and the goal is to describe the actual prosodic realizations from speakers. See the work of [1, 2, 3], where the first reference offers a good review of the field.

The most restricted case of looking at symbolic prosody assignment from text alone –the case of importance to us because of its relevance to speech synthesis– has received comparatively far less

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

attention, with some early work looking at the use of decision trees (DT) for classification of prominence and boundaries [4], and DTs coupled with Markov chains to model the temporal evolution of these categories [5, 6]. None of this work explores the type of lexical representations we are interested in.

For our work, we rely on various previously published lexical-embedding approaches, focusing on the application of two types of models for symbolic-prominence prediction: the Word-to-Vector [7, 8] and Global Vector [9] models, which will be described in more detail in Section 3. The lexical-embedding literature is rife with recent developments and includes other noteworthy variants, such as embeddings exploiting dependency-parse information [10, 11] (which we do not explore here).

The application of embeddings to symbolic-prominence prediction in a TTS front-end is a novel contribution of this paper. However, other authors have looked at related ideas for synthesis, such as the use of embeddings in the direct prediction of vocoder parameters in a parametric synthesizer [12]. The most closely-related work to ours is that of [13] where the authors construct word embeddings for phrase-break predictions from the first layer of a neural network that updates the embeddings as it trains the phrase-break predictor. Our work differs, however, in the type of embeddings used, in the use of recurrent architectures, in the extension to prominence modeling, and in the use of a two-stage approach (we use embeddings derived from large amounts of unsupervised data, and then deploy them in a supervised-learning task).

3. OVERVIEW OF WORD-EMBEDDING MODELS

In this section we provide a brief overview of the embedding models explored in this work: (i) three formulations of the word-to-vector embedding, an embedding technique based on the use of a context window (either at the input or at the output), and (ii) the Global Vector (GloVe) embedding, which on the other hand derives embeddings by taking into account global co-occurrence counts over a corpus.

3.1. Word-to-Vector: Skip-gram

The skip-gram model introduced in [7, 8] can be represented as a neural network with a single hidden layer, in which a V-dimensional one-hot encoded word input ($\mathbf{w}_I$) is used to simultaneously predict C words of context ($\mathbf{w}_{O,1}, \cdots, \mathbf{w}_{O,C}$), as shown in Fig. 1 (a).

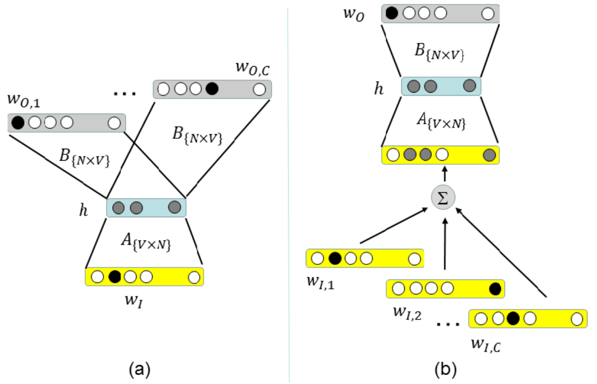

Fig. 1. *Word-to-Vector Models: (a) Skip-Gram, and (b) Continuous Bag of Words (CBOW)*

Here $A_{V\times N}$ is the input-to-hidden matrix linearly mapping input words to the N-dimensional embedded space, and $B_{N\times V}$ is a single hidden-to-output matrix that is shared by all the output words and allows the reconstruction of the context words from the embedding of the input word. Since $\mathbf{w}_I$ is a one-hot vector, this means that the product

$$\mathbf{h}^T = \mathbf{w}_I^T A = A_{(k,:)} \tag{1}$$

simply picks the k-th row of A whenever $\mathbf{w}_I = k$. The rows of the A matrix therefore contain, once the model has been properly trained, the desired N-dimensional embeddings for each word in the vocabulary. The network can be trained to minimize the negative log-likelihood of the output context given the input word. Modeling each of the C output context *panels* with multinomial distributions, it can be shown this is equivalent to minimizing the following criterion:

$$\begin{aligned} E_{SG} &= -\log p(\mathbf{w}_{O,1}, \cdots, \mathbf{w}_{O,C}|\mathbf{w}_I) \\ &= -\sum_{c=1}^{C} u_{j_c^*} + C \log \sum_{j'=1}^{V} \exp(u_{j'}), \end{aligned} \tag{2}$$

where $\mathbf{u} = B^T\mathbf{h}$ is the vector of activations arriving at each of the C output panels, and $j_c^* \in [1, \cdots, V]$ is the vocabulary index of the actual c-th output context word.

3.2. Word-to-Vector: Continuous Bag of Words (CBOW)

The Continuous Bag of Words (CBOW) word-to-vector model inverts the formulation of the skip-gram model, attempting to predict the current word $\mathbf{w}_O$ based on a multi-word context $\mathbf{w}_{I,1}, \cdots, \mathbf{w}_{I,C}$, as shown on Fig. 1 (b). As in the skip-gram model, we assume the inputs and outputs are orthogonal, hot-vector representations of a V-sized vocabulary, with the A and B matrices retaining the same interpretations as in the previous model. In this model, however, the input-to-hidden matrix is applied to the *average* input vector:

$$\mathbf{h}^T = \frac{1}{C}(\mathbf{w}_{I,1} + \cdots + \mathbf{w}_{I,C})^T A = \frac{1}{C}\sum_{c=1}^{C} A_{(k_c,:)}, \tag{3}$$

where now the multiplication by the (scaled) sum of input binary vector causes an averaging of the rows of A (containing each word's embeddings). The loss function can be similarly defined as in the case of the skip-gram model to be the negative log likelihood of a multinomial (modeled with a softmax) over the V output units:

$$\begin{aligned} E_{CBOW} &= -\log p(\mathbf{w}_O|\mathbf{w}_{I,1}, \cdots, \mathbf{w}_{I,C}) \\ &= -u_{j_O^*} + \log \sum_{j'=1}^{V} \exp(u_{j'}) \end{aligned} \tag{4}$$

where $\mathbf{u} = B^T\mathbf{h}$ is the softmax argument, and j_O^* is the index of the output word.

3.3. Word-to-vector: Structured Skip-gram

The structured skip-gram model is introduced in [14] as a modification of the skip-gram model where instead of a single hidden-to-output matrix $B_{N\times V}$ that is shared by all the output words, there is a distinct matrix for each of the context words. That way, the model is sensitive to the positioning of the context words, and may be more suitable for syntax-related tasks.

3.4. GloVe

The GloVe (Global Vector) model [9] is a simple, co-occurrence-based approach that seeks to construct continuous vector embeddings of words, so as to minimize the (weighted) reconstruction, in a least-squares sense, between the dot product of anchor and context words, and their log co-occurrence counts. Specifically, the embeddings can be obtained by minimizing the following criterion function:

$$J = \sum_{i,j=1}^{V} f(X_{ij})\big(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij})\big)^2, \tag{5}$$

where X_{ij} is the number of times word j appears in the context of word i for a predefined context window of length L; w and $\tilde{w}$ are the word and context-word vectors (and, analogously, b and $\tilde{b}$ the respective biases), and $f()$ is the piecewise weighting function:

$$f(X_{ij}) = \begin{cases} (X_{ij}/X_{max})^{\alpha} & \text{if } X_{ij} < X_{max} \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

As we describe in Section 5 when reporting various experiments, we have relied on GloVe embeddings of various dimensionality, pre-trained on various corpora, and which are publicly available from [15]. The authors report computing these embeddings with the following choice of hyper-parameters for the above criterion: $X_{max} = 100$, $\alpha = 3/4$, and $L = 10$ [9].

4. SYMBOLIC PROSODY PREDICTION

As a modeling approach we adopt Bidirectional Recurrent Neural Networks (BiRNN) that use Long Short-Term Memory (LSTM) units as hidden-layer non-linearities. BiRNN/LSTM models have been recently shown to provide state-of-the-performance across various dynamic modeling tasks that involve complex contextual dependencies, including various prosodic regression and classification tasks (e.g., prosodic-contour modeling [16, 17, 18, 19], phrase-break prediction [20], and labeling of audio corpora with phrasing and prominence labels [3]). The LSTM non-linearities, which act as memory cells within the network, have been shown to be fundamental to alleviate training issues such as vanishing gradients. The use of bidirectional structures allow contextual information from both past and future inputs to influence the prediction at every time step, which is a desirable property for tasks such as prominence placement where upcoming word tokens can shape the speaker's choice for the current word. Additionally, as in the case of feed-forward networks, several layers of BiRNN/LSTM models can be stacked to create compositionally deep models. In the interest of space, and given recent treatment elsewhere, the reader is directed to consult one of the references provided for more details of the BiRNN/LSTM models, including forward-pass equations (see, e.g., [16]). To train the models for the experiments reported here, we have relied on the Theano toolkit to provide the computation of the necessary gradients using back-propagation through time, and weight updates using stochastic gradient descent [21].

4.1. Baseline Feature Set

In this section we describe the set of features that serve as inputs to the baseline system that we wish to improve upon with the use of additional lexical embeddings. These are all textual features that have been previously reported in the literature for various types of prosodic modeling, and they are defined for each word w_i as follows:

- the part-of-speech (POS) tag
- the normalized pointwise mutual information (NPMI) with respect to the previous and following words: $NPMI(w_i, w_{i-1})$ and $NPMI(w_i, w_{i+1})$, where:

$$NPMI(x, y) = \log \frac{p(x)p(y)}{p(x, y)} / \log p(x, y) \tag{7}$$

- the unigram probability $p(w_i)$ (this, and the probabilities needed to estimate NPMI in Eq. 7, are obtained from a smoothed 4-gram language model)
- type of following punctuation
- the following 6 boolean features indicating whether w_i is: a) capitalized, b) an adposition, c) a conjunction, d) an auxiliary verb, e) a WH word, f) a function word.
- the following 2 parse features indicating syntactical coupling: let N be the shallowest node in a parse tree dominating terminals (words) w_i and w_{i+1}, and let d_i and d_{i+1} be their respective distance (i.e., number of intermediate nodes) to N. Define then $\min(d_i, d_{i+1})$ and $\max(d_i, d_{i+1})$ as features.
- the Pitch-Accent Ratio (PAR): this feature provides a solid baseline lexical feature that has already been investigated for prominence modeling after being introduced in [22]. The PAR is a context-free, memory-based feature summarizing prior knowledge about the pitch-accentability, or probability of a word being prominence-bearing. It is based on the fraction of the time the unigram receives prominence in a labeled corpus, subject to a statistical significance test, and is given by:

$$PAR = \begin{cases} \frac{PRM_w}{N_w} & \text{if } B(PRM_w, N_w; \theta) \leq 0.05 \\ 0.5 & \text{otherwise,} \end{cases} \tag{8}$$

  where N_w is the number of times a word w appears in the corpus, PRM_w the number of times that it is prominent, and $B(PRM_w, N_w; \theta)$ is a binomial distribution with parameter $\theta = 0.5$. This feature equals the fraction of "successes" whenever there is sufficient evidence in a corpus to establish how likely a word co-occurs with a prominence event, as diagnosed by a binomial distribution. With insufficient evidence, the feature reflects uncertainty and is set to 0.5.

A baseline feature vector is constructed from all the previous features, after encoding any categorical features as binary hot vectors.

5. EVALUATION

For evaluating the different approaches, we relied on a speech synthesis corpus of professionally recorded speech from a female native speaker of North American English. 3730 sentences from this corpus have been fully annotated with the Tones and Break Indices (ToBI) inventory by an expert annotator who had access to the audio and their text transcripts. The different pitch-accent labels of the inventory were used to derive binary prominence labels for this work. For the prosodic phrasing task, we consider intonation phrase boundaries with index '4'. Only the text was used to derive the input features, and the labels matching the actual prosodic realizations of the speaker were used as ground truth for training and evaluation. The utterances of the corpus were split into disjoint training (80%), development (10%) and test (10%) sets, resulting in approximately

47.8K, 6.3K, and 5.9K word tokens respectively. The development set was used for diagnosing convergence during the RNN training. The training set was used for training the models, and for tabulating the Pitch-Accent Ratio feature of Eq. 8.

A 3-layer neural network model consisting of the following structure was trained: a single non-recurrent layer (with 160 hyperbolic tangent activation units), followed by 2 stacks of bidirectional layers, each with 80x2 LSTM hidden units, and a binary output softmax layer. Since LSTM layers dominate the number of parameters, we use a non-recurrent layer as the first layer, so that input features of various dimensionalities (for the various experiments considered) produce models of roughly comparable sizes (in the range of 320K to 370K parameters).

The following models were trained:

- BL: A baseline model including the baseline features described in Section 4.1 only.
- LEX-100 (LEX-300): A model using the baseline features, augmented with a one-hot vector of word identities corresponding to the most frequent 100 (300) words in the Gigaword corpus. This provides an alternative baseline to the embedded models using raw lexical features.
- SG-50: A model using the baseline features augmented with 50-dimensional embeddings from a skip-gram model we trained on the Gigaword corpus [23], with an output layer containing a context window with the previous and following 5 words (using the tools available in [24]).
- SSG-50 (SSG-100): A model using the baseline features augmented with 50-dimensional (100-dimensional) embeddings from a structured-skip-gram model we trained on the Gigaword corpus, with an output layer containing a context window with the previous and following 5 words (using the tools available in [14]).
- CBOW-300: A model using the baseline features augmented with pre-trained, 300-dimensional embeddings from a CBOW model. The embeddings used in this experiment were trained the GoogleNews corpus, and are available from [24].
- GloVe-50: A model using the baseline features augmented with pre-trained, 50-dimensional embeddings from a GloVe model. These embeddings were derived from approximately 6 billion word tokens from Wikipedia and Gigaword, and are available from [15].
- GloVe-300: A model using the baseline features, augmented with pre-trained, 300-dimensional embeddings from a GloVe model. These embeddings were trained on about 840 billion word tokens from the Common Crawl [25], and are also available from [15].

For all the embedding experiments, we used a vocabulary of 70K words. In the case of models trained with pre-published embeddings (CBOW-300, GloVe-50, and GloVe-300), we extracted the embeddings relevant to our target vocabulary. For all the experiments using embedding features, we assigned the zero vector to any out-of-vocabulary words. Prior to training, all continuous-valued input features were z-scored with respect to the training-set mean and standard deviation (binary features are left intact). For each of the models defined above, 10 different systems were trained using different random-seed initialization for the weights and mini-batch allocation. The test-set average precision, recall, and $F1$ scores over all 10 systems are reported in Tables 1 and 2 (with respective standard deviations in parenthesis).

Table 1. *Metrics for the prominence prediction task for the baseline model and models using embedding features.*

Model	Precision	Recall	F1
BL	0.644 (0.017)	0.444 (0.034)	0.525 (0.019)
LEX-100	0.649 (0.009)	0.458 (0.013)	0.537 (0.009)
LEX-300	0.683 (0.014)	0.457 (0.025)	0.547 (0.016)
SG-50	0.706 (0.012)	0.600 (0.033)	0.648 (0.016)
SSG-50	0.708 (0.012)	0.583 (0.037)	0.639 (0.019)
SSG-100	0.718 (0.012)	0.627 (0.024)	0.669 (0.011)
GloVe-50	0.712 (0.017)	0.599 (0.030)	0.650 (0.012)
GloVe-300	0.748 (0.011)	0.673 (0.022)	**0.708 (0.009)**
CBOW-300	0.742 (0.021)	0.686 (0.033)	**0.712 (0.011)**

Table 2. *Metrics for the intonational phrase break prediction task for the baseline model and models using embedding features.*

Model	Precision	Recall	F1
BL	0.827 (0.014)	0.796 (0.010)	0.811 (0.004)
LEX-100	0.834 (0.008)	0.796 (0.006)	0.815 (0.004)
LEX-300	0.834 (0.016)	0.805 (0.019)	0.819 (0.005)
SG-50	0.836 (0.013)	0.807 (0.018)	0.821 (0.005)
SSG-50	0.839 (0.015)	0.814 (0.014)	0.826 (0.002)
SSG-100	0.841 (0.015)	0.820 (0.020)	**0.830 (0.004)**
GloVe-50	0.837 (0.013)	0.809 (0.012)	0.823 (0.004)
GloVe-300	0.841 (0.014)	0.809 (0.017)	0.824 (0.004)
CBOW-300	0.834 (0.016)	0.815 (0.015)	0.824 (0.003)

6. DISCUSSION AND CONCLUSIONS

In this work we have investigated the use of continuous lexical representations as features in prosodic phrasing- and prominence- prediction tasks, such as one encounters in the text-processing front-end of a TTS system. We have explored the contribution of embeddings from three Word-to-Vector formulations and the GloVe model, and shown that in all cases (and particularly for the top 2 models) the alternatives outperform both a baseline model which included no explicit word identity information, as well as two naïve baselines (LEX models) which included raw word identity features. It is worth noting that since in this naïve lexical model the size of the input grows with the size of the dictionary, it does not scale well and can suffer from data sparsity during training. As we see, it does not significantly outperform the basic baseline, further motivating the use of the embedded features for this task. (The minor difference between the LEX-100 and LEX-300 models, however, suggests this model could learn better given a larger input vocabulary and more training data). Although more exploration is needed, our experiments seem to suggest that the choice of embedding dimensionality is more crucial to the prediction task than the particular choice of embedding model, with the highest absolute gains of approximately 0.187 in F1 score (35% relative gain) being obtained with 300-dimensional embeddings on the prominence prediction task. For the phrasing-prediction tasks, the improvements are only modest, with the best result (0.019 gain in F1 score, or 2.4% relative) achieved with the structured-skip-gram embedding model, as this task is likely more related to syntax. Future work in this area includes exploring other embeddings to see how stable these results are across models and dimensions and how the phrasing-break prediction may be improved, as well as testing the technique on other corpora and genres.

7. REFERENCES

[1] A. Rosenberg, *Automatic Detection and Classification of Prosodic Events*, Ph.D. thesis, Columbia University, 2009.

[2] R. Fernandez and B. Ramabhadran, “Driscriminative training and unsupervised adaptation for labeling prosodic events with limited training data,” in *Interspeech*, Tokyo, Japan, Sept. 2010, pp. 1429–1432.

[3] A. Rosenberg, R. Fernandez, and B. Ramabhadran, “Modeling phrasing and prominence using deep recurrent learning,” in *Interspeech*, Dresden, 2015, pp. 3066–3070.

[4] J. Hirschberg, “Pitch accent in context: Predicting intonational prominence from text,” *Artificial Intelligence*, vol. 63, pp. 305–340, 1995.

[5] K. Ross and M. Ostendorf, “Prediction of abstract prosodic labels for speech synthesis,” *Computer Speech & Language*, vol. 10, no. 3, pp. 155–185, 1996.

[6] I. Read and S. Cox, “Automatic pitch accent prediction for text-to-speech synthesis,” in *Interspeech*, Antwerp, 2007, pp. 297–300.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proc. ICLR*, 2013.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. NIPS*, 2013, pp. 3111–3119.

[9] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 2014, pp. 1532–1543, Association for Computational Linguistics.

[10] M. Bansal, “Dependency link embeddings: Continuous representations of syntactic substructures,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, June 2015, pp. 102–108, Association for Computational Linguistics.

[11] O. Levy and Y. Goldberg, “Dependency-based word embeddings,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, vol. 2, pp. 302–308.

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

[13] A. Vadapalli and K. Prahallad, “Learning continuous-valued word representations for phrase break prediction,” in *Proc. Interspeech*, Singapore, 2014, pp. 41–45.

[14] “wang2vec: Extension of the original word2vec using different architectures,” `http://github.com/wlin12/wang2vec`, 2015, [Online; accessed Sept-2015].

[15] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” `http://nlp.stanford.edu/projects/glove/`, 2015, [Online; accessed Sept-2015].

[16] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, “Prosody contour prediction with long short-term memory, bidirectional, deep recurrent neural networks,” in *Interspeech*, Singapore, 2014, pp. 2268–2272.

[17] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, “Using deep bidirectional recurrent neural networks for prosodic-target prediction in a unit-selection text-to-speech system,” in *Interspeech*, Dresden, 2015, pp. 1606–1610.

[18] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, “TTS synthesis with bidirectional LSTM based recurrent neural networks,” in *Interspeech*, Singapore, 2014, pp. 1964–1968.

[19] H. Zen and H. Sak, “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis,” in *ICASSP*, Brisbane, 2015, pp. 4470–4474.

[20] A. Rosenberg, R. Fernandez, and B. Ramabhadran, “Phrase boundary assignment from text in multiple domains,” in *Interspeech*, Portland, 2012, pp. 2558–2561.

[21] 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.

[22] A. Nenkova, J. Brenier, A. Kothari, S. Calhoun, L. Whitton, D. Beaver, and D. Jurafsky, “To memorize or to predict: Prominence labeling in conversational speech,” in *Proc. HLT-ACL*, Rochester, NY, April 2007, pp. 9–16.

[23] D. Graff and C. Cieri, “English Gigaword,” `http://catalog.ldc.upenn.edu/LDC2003T05`, 2003.

[24] “word2vec: Tool for computing continuous distributed representation of words,” `http://code.google.com/p/word2vec/`, 2015, [Online; accessed Sept-2015].

[25] “Common crawl,” `http://https://commoncrawl.org/`, [Online].