Articulatory Movement Prediction Using Deep Bidirectional Long Short-Term Memory Based Recurrent Neural Networks and Word/Phone Embeddings

Pengcheng Zhu[1], *Lei Xie*[1,2], *Yunlin Chen*[1]

[1]School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an, China
[2]School of Computer Science, Northwestern Polytechnical University, Xi'an, China

{pczhu, lxie, ylchen} @nwpu-aslp.org

Abstract

Automatic prediction of articulatory movements from speech or text can be beneficial for many applications such as speech recognition and synthesis. A recent approach has reported state-of-the-art performance in speech-to-articulatory prediction using feed forward neural networks. In this paper, we investigate the feasibility of using bidirectional long short-term memory based recurrent neural networks (BLSTM-RNNs) in articulatory movement prediction because they have long-context trajectory modeling ability. We show on the MNGU0 dataset that BLSTM-RNN apparently outperforms feed forward networks and pushes the state-of-the-art RMSE from 0.885 mm to 0.565 mm. On the other hand, predicting articulatory information from text heavily relies on handcrafted linguistic and prosodic features, e.g., POS and TOBI labels. In this paper, we propose to use word and phone embeddings to substitute these manual features. Word/phone embedding features are automatically learned from unlabeled text data by a neural network language model. We show that word and phone embeddings can achieve comparable performance without using POS and TOBI features. More promisingly, combining the conventional full feature set with phone embedding, the lowest RMSE is achieved.

Index Terms: articulatory movement predictions, articulatory inversion, long short term memory (LSTM), word2vec, recurrent neural network (RNN)

1. Introduction

Human speech originated from articulatory movements that involve systematic combinations of motions from tongue, jaw, lips, velum, etc. These movements can be accurately recorded by human articulography, e.g., electromagnetic articulography (EMA) [1]. But it often needs a cumbersome setup and a complicated recording procedure. Automated systems capable of approximating the position of the articulators from acoustic speech or text are known to be quite useful in many practical applications. In speech recognition, articulatory information can provide additional speech production knowledge to improve the recognition performance [2,3]. In speech synthesis, articulatory information is used to improve the voice quality or to modify the characteristics of the synthesized voice [4, 5]. In audio-visual speech processing, articulatory features can be regarded as an intermediate parametrization of speech that has close link with facial feature positions. Hence articulatory features can be used to synthesize natural facial animation for language tutoring or natural user interface [6, 7].

1.1. Related Works and Problems

Many methods have been previously proposed to predict articulatory movements [5, 8–11]. When acoustic features are the input to estimate the articulatory movements, the problem is also known as *acoustic-to-articulatory mapping* or *articulatory inversion*. In [5], a Gaussian mixture model (GMM) for the joint distribution of acoustic and articulatory features was adopted to achieve the mapping from acoustic features to articulatory features. In [9], a hidden Markov model (HMM) approach to articulatory movement prediction from speech was presented, which adopted a similar framework to HMM-based parametric speech synthesis. There are several variants of the HMM framework [8, 9, 12, 13] and HMMs are also proven to be quite useful in articulatory prediction from text [8]. Similar to text-to-speech (TTS), HMM-based articulatory prediction usually adopts a rich set of features, including linguistic and prosodic representations. In [8], combination of text and acoustic features led to further gain in prediction accuracy. The articulatory movement prediction problem can be directly evaluated by prediction error metrics like root mean squared error (RMSE), while a recent research from [14] has studied the task-specific evaluation method.

Artificial neural networks (ANNs) have been proven to be quite effective in regression tasks like acoustic-to-articulatory mapping [10, 15–17]. In an early study [18], a multilayer perceptions (MLP) approach has been used and the predicted articulatory movements have proved to be quite useful for continuous speech recognition. Later, Richmond proposed a trajectory mixture density network (TMDN). With the help of the maximum-likelihood parameter generation (MLPG) algorithm, his approach is able to provide smooth articulatory trajectories. Recently, neural networks with multiple hidden layers, i.e., deep neutral networks (DNNs), have achieved tremendous success in speech recognition [19, 20] and synthesis [21, 22]. To the best of our knowledge, Uria [23] is the first one who introduced deep networks into the articulatory inversion task. Specifically, he investigated a DNN and a deep version of the TMDN and obtained an average RMSE of 0.885 mm on the MNGU0 test dataset [1]. As far as we know, this is the state-of-the-art inversion accuracy publicly ever reported. However, in order to model the temporal context information of speech, he borrowed the bigger feature window idea usually used in DNN-HMM speech recognition [24]. That is, a pre-defined fixed-length context window, covering several frames of acoustic features, is used as the network input. Moreover, this approach only uses piecewise projections to estimate articulatory movements. The temporal correlations in the whole speech utterance are apparently neglected.

On the other hand, predicting articulatory information from text can be problematic. Current approaches highly rely on rich linguistic and prosodic features, such as part-of-speech (POS) labels and tone and break index (TOBI). Manually labeling these features is quite expensive. Most importantly, annotation needs particular human expertise and tremendous efforts. Machine learning methods can be used to predict these features, but the performance is far from satisfactory, especially for TOBI labeling [25]. Even worse, prediction errors will definitely spread to the downstream articulatory prediction step.

1.2. The Proposed Approach

In this paper, we propose a new approach to (1) model the long-range speech dynamics more precisely through a recurrent neural network (RNN) and (2) substitute the manual POS and TOBI features with automatically learned word/phone embeddings through a neural network. Specifically, to achieve (1), we use a deep bidirectional long-short term memory (BLSTM) based RNN to model the speech/text to articulatory mapping. This is inspired by the recent success of its long-context trajectory modeling ability in speech recognition and synthesis. Moreover, different from previous studies in which line spectral frequencies (LSF) are used as acoustic features, we use MFCC as input in articulatory inversion. Experiments on MNGU0 dataset show that the BLSTM-RNN with MFCC input pushes the state-of-the-art performance in articulatory inversion from 0.885 mm to 0.565 mm in term of RMSE. To achieve (2), we use neural network based word embedding [26, 27] and phoneme embedding as features. This is inspired from the successful use of word embedding as POS and TOBI feature substitution in a recent TTS approach [28]. Word embedding is a low dimensional continuous-valued vector effectively used to represent a word. This feature, learned from unlabelled text data in a fully unsupervised way, is assumed to carry important syntactic and semantic information [29]. More importantly, we propose *phone embedding* that is learned from triphone sequences simply converted from text. Interestingly, we discover that the obtained triphone vectors convey pronunciation similarity information. Experiments on text-to-articulatory mapping show that word and phone embeddings can achieve comparable performance without using POS and TOBI features and promisingly, combing conventional full feature set with phone embedding, the lowest RMSE is achieved.

2. Network Architecture

A recurrent neural network (RNN) is a typical class of neural network in which connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. In an RNN, given an input sequence $\boldsymbol{x} = [x_1, ..., x_T]$, the hidden vector $\boldsymbol{h} = [h_1, ..., h_T]$ and the output vector $\boldsymbol{y} = [y_1, ..., y_T]$ can be computed from $t = 1$ to T according to:

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \quad (1)$$

$$y_t = W_{hy}h_t + b_y, \quad (2)$$

where $\mathcal{H}$ is the activation function of a hidden layer, W terms denote the weight matrices and the b terms are the bias vectors.

In order to fully make use of the context of input sequences in both preceding and succeeding directions, bidirectional RNNs (BRNNs) have been proposed [30]. As shown in Fig. 1, BRNNs compute the forward sequence $\overrightarrow{h}$ and the backward sequence $\overleftarrow{h}$ by iterating the forward layer from $t = 1$ to T with the following iterating functions.

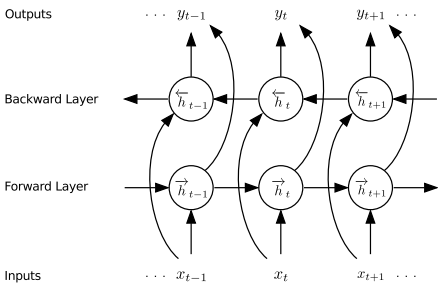

Figure 1: Bidirectional recurrent neural netwok.

Figure 2: Long short-term memory cell.

$$\overrightarrow{h}_t = \mathcal{H}(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}), \quad (3)$$

$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}), \quad (4)$$

$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y. \quad (5)$$

In a standard RNN, $\mathcal{H}$ is usually a sigmoid or hyperbolic tangent function, which leads to the limitation of the inability to learn long-range context dependencies. A network with long short-term memory (LSTM) blocks can solve this problem. An LSTM network consists of recurrently connected blocks, known as memory blocks. The structure of a single LSTM memory block is illustrated in Fig. 2. Every memory block contains self-connected memory cells and three adaptive and multiplicative gate units (input, output and forget gates), which can respectively provide writing, reading and resetting operations for the cells. Among them, forget gates are shown to be essential for problems involving continual or very long strings [31].

Combining the advantages of BRNNs and LSTMs, bidirectional LSTM based RNNs have been designed [32], which can make use of long-range context in both forward and backward directions. Motivated by the recent success of deep network architectures, deep BLSTM-RNNs are considered to build up high level representations of input features. BLSTM-RNNs have been successfully used in regression tasks like speech synthesis [28] and visual speech synthesis [33]. In this paper, we introduce BLSTM-RNNs into the task of articulatory movement prediction with speech and text input.

3. Word/Phone Embedding

Previous text-to-articulatory-movement prediction has made use of a broad set of hand-crafted linguistic and prosodic features [34], including part of speech (POS) tags and tones and breaks indices (TOBI). These features are usually manually labelled or predicted using machine learning methods. We aim to use neural network based word and phone vector represen-

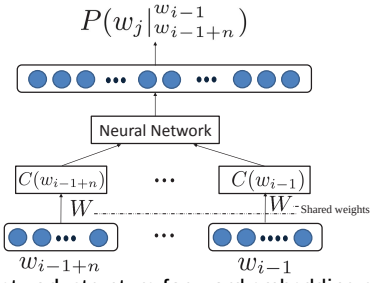

Figure 3: Network structure for word embedding and NNLM.

tations automatically learned from unlabeled text data to substitute these manual features. Previous studies have shown that neural network based word vectors encode many linguistic regularities and semantics that can be used as features in natural language processing applications [35]. A recent study has shown that word embedding can serve as a useful feature in TTS synthesis [28].

3.1. Word/Phone Embedding Generation

Word embedding can be learned by a neural network language model (NNLM) which predicts the current word's probability distribution from previous words. In order to train an NNLM, a large text corpus is used, which consists of a large number of sentences represented by word sequence $w_1, w_2, \cdots, w_t, \cdots, w_T$, where $w_t \in V$ and V is a large and finite vocabulary set. The objective is to learn a model [36]:

$$f(w_t, ..., w_{t-n+1}) = P(w_t|w_1^{t-1}). \quad (6)$$

To achieve this, the function is decomposed into two parts: a mapping C from any word w_i in V to a real-valued vector $C(w_i)$ and a function g mapping an input sequence of word with vector representations $(C(w_{t-n+1}), \cdots C(w_{t-1}))$ to a conditional probability distribution over words in V for the next word w_t. The output of g is a vector of size V and its ith element estimates the probability $P(w_t = i|w_1^{t-1})$. So we have

$$f(i,w_{t-1},\cdots,w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1})). \quad (7)$$

Hence the original function f is composed of two mappings, C and g, with C being shared across all the words in the context. C is a $|V| \times m$ matrix whose row i is the vector representation $C(w_i)$ for word w_i. In practice, all words are represented with one hot representation (1-of-V) whose dimension is $|V|$. Each input word vector w_i is mapped to $C(w_i)$ which has a much lower dimension m by multiplying a weight matrix $W(m \times |V|) : C(w_i) = Ww_i$. The two mappings ($C$ and g) are realized by neural networks and trained by back propagation. The main difference between training W and training the weights of mapping g is that all input words share one W. The network structure is shown in Fig. 3.

Currently, such kind of vector representation is only limited to the word level. We apply the above method to a lower granularity, i.e., phone level. First, we expand a word sequence $w_1, w_2, \cdots, w_t, \cdots, w_T$ to its triphone sequence $p_1, p_2, \cdots, p_t, \cdots, p_{T'}$ by cross-word expansion according to a pronunciation dictionary. Then using the method above, we obtain the vector representation $C(p_i)$ for a triphone p_i.

In our study, *word2vec*[1] from Google is used, which is a tool for efficient implementation of the continuous bag-of-words (CBOW) and skip-gram architectures for computing vector representations of words. Specifically, we use CBOW architecture in our implementation. Please refer to [37] for more details.

[1]https://code.google.com/p/word2vec/

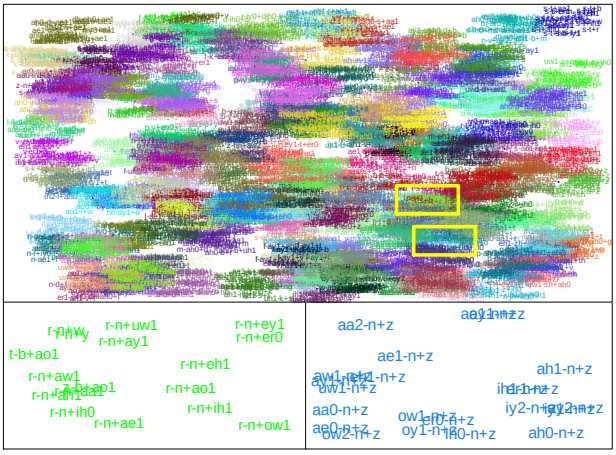

Figure 4: Clustering results on triphone vectors

3.2. Meanings of Phone Embedding

Previous approaches have shown that word embedding carries a certain syntactic and semantic information. For example, if we perform clustering on the word vectors, "meaningful" word categorises are generated. Words that are similar in semantics, e.g., car, automobile, van, BMW, Ford, are clustered to one category. Hence we are interested to see what kind of information phone level embedding carries. Similarly, we perform k-means clustering on the triphone vectors (k=500) and the clustering results are shown in Fig. 4. The t-SNE algorithm [38] is used for dimension reduction of the representations, so as to generate the two-dimensional plots for visualization. We interestingly discover that triphones are clustered according to their pronunciation similarity in general. In Fig. 4, two clusters are zoomed in and about 30 samples are shown for each cluster. For the cluster on the left (green), we can see that for most samples, the central phone is /t/, the left context is /r/ and the right context is a vowel. For the cluster on the right (blue), we can observe that the central vowel is /n/, the right context is /z/ and the left context is a vowel. Since the triphones in each cluster are kind of similar in pronunciations, the articulatory movements producing these pronunciations should be similar too. We believe that triphone vectors represent the pronunciation similarity information to some extent.

4. Experiments

4.1. Experimental Setup

Our experiments were carried out on MNGU0 [1] database with 1, 263 English utterances from a single speaker in a single session. Parallel recordings of acoustic data and EMA data are available. EMA data are collected with a sampling frequency of 200Hz from 6 sensors located at the *tongue dorsum* (T3), *tongue body* (T2), *tongue tip* (T1), *lower lip* (LL), *upper lip* (UL), and *lower incisor* (LI). We exactly followed the experimental configurations in previous studies [23]. Only x- and y-coordinates of the 6 receivers were used in the experiments because the movements in z-axis were very small. The acoustic feature consists of 40 frequency warped line spectral frequencies (LSFs) and a gain value and the frame shift step is 5 ms in order to obtain acoustic features at the same frequency as the EMA data. The database is partitioned into three sets: validation and test sets comprising 63 utterances each, and a training set consisting of the other 1, 137 utterances.

The word and phone vectors were trained using the English

Table 1: *Results for networks with different hidden layers and number of nodes. (B: bidirectional RNN; F: Feed-forward)*

Nodes	BBB	BBF	BFB	BFF	FBB	FBF	FFB	FFF
64	1.127	1.299	0.960	1.210	0.984	1.059	1.061	1.325
128	0.970	1.317	0.964	1.201	**0.889**	0.971	1.014	1.482
256	1.041	1.284	1.125	1.184	0.901	1.264	0.993	1.542

Table 2: *Performance comparison between LSF and MFCC.*

Feature/Node	64	128	256
LSF	0.984	0.889	0.901
MFCC	0.599	**0.565**	0.585

wikipedia text data [2] (95.3M). The vocabulary size is about 256K and 81K for words and triphones, respectively. We embedded both word and phone into a 100-dimensional vector. A popular toolkit named CURRENT was used for neural network training. We set the learning rate and the momentum to 1e-6 and 0.9, respectively and the weights were initialized with a Gaussian random distribution. The training procedure stops when the sum square error on the validation set no longer declines within the last 10 epochs. We conducted evaluations by directly comparing the predicted articulatory movements with the original EMA data. The commonly used error metric, root mean-squared error (RMSE), was used for objective evaluation.

4.2. BLSTM-RNNs for Articulatory Inversion

We tested the articulatory inversion performance of a set of network topologies with different hidden layers (F: feed forward, B: BLSTM) and node sizes (64, 128, 256). Results show that the 3-hidden-layer structures outperform the 1- and 2-hidden layer structures in general. The results for the tested 3-hidden-layer structures are summarized in Table 1. We interestingly found that, the topologies with a bidirectional layer (B) performs consistently better than those with only feed forward (F) layers. The best performed network topology is the one with two BSLTM layers sitting on top of one feed-forward layer (FBB). We keep this structure and further adjust the number of nodes in each hidden layer. The best performance is achieved by a 150-node network, with the lowest RMSE of 0.867mm, which outperforms the state-of-the-art performance (0.885mm) achieved by a deep trajectory mixture density network (DTMDN) [23].

4.3. LSF vs. MFCC

In the last decade, most studies on articulatory inversion took line spectral frequencies (LSF) as the acoustic feature. This is reasonable because LSF is an articulatory-originated feature. Interestingly, we discovered that MFCC is able to boost the prediction error to a new low level. Framewise 39-dimensional MFCC features were extracted at the same frame rate of LSF. Performance comparison is summarized in Table 2, in which the network structure is FBB. We can clearly find that the RMSE achieved by MFCC is much lower than LSF. The FBB network with 128 hidden nodes each layer achieves the lowest RMSE of 0.565. This is a new record in articulatory inversion experiments conducted on the MNGU0 dataset.

4.4. Word/Phone Embeddings

We tested text-to-articulatory-movement prediction using an FBB128 network. We split the input features into four subsets:

- **Basic** (321 Dim): the broad linguistic context feature set from Table 1 of [8], excluding POS and TOBI;
- **POS&TOBI** (35 Dim): the POS and TOBI features from Table 1 of [8];

[2] http://mattmahoney.net/dc/enwik9.zip

Table 3: *Results for text-to-articulatory prediction.*

Features	RMSE
Basic + POS&TOBI	1.870
Basic	1.925
Word2vec	2.530
Triphone2vec	2.348
Basic+Word2vec	1.894
Basic+Triphone2vec	1.881
Basic+POS&TOBI+Word2vec	1.782
Basic+POS&TOBI+Triphone2vec	**1.734**

- **Word2vec** (100 Dim): the word vector feature introduced in Section 4.1;
- **Triphone2vec** (100 Dim): the triphone vector feature introduced in Section 4.1.

Results are listed in Table 3. First, we notice that the RMSE remains at a high level compared with acoustic-to-articulatory prediction. The same observation is also reported in [8]. When POS and TOBI features are removed (only Basic), the RMSE has a notable increase. This shows the importance of these handcrafted features. When Word2vec feature is added to the basic feature set, the RMSE drops to a comparable value with the full feature set (Basic + POS&TOBI). The addition of triphone2vec to the Basic feature set shows a more promising result with a lower RMSE much closer to Basic+POS&TOBI. When Word2vec or Triphone2vec is further combined with Basic+POS&TOBI, interestingly we observe obvious decrease in RMSE. The lowest RMSE is achieved by Basic+POS&TOBI+Triphone2vec with 10% and 7.3% relative RMSE reduction compared with Basic and Basic+POS&TOBI, respectively. This may indicate that the word/phone embedding features are complimentary with POS and TOBI features in the articulatory prediction task. We believe that the superior performance gain achieved by triphone2vec may come from its embedded pronunciation similarity as discussed in Section 3.2.

5. Conclusions and Future Work

Our contributions are two fold. First, we boost the articulatory inversion performance to a new level by the use of BLSTM-RNN and the MFCC feature input. The best RMSE reported before is 0.885mm in [23] and our approach achieves 0.565mm. Second, in text-to-articulatory-movement-prediction, we manage to substitute POS and TOBI features with neural network based word and phone vector features that are automatically learned from unlabelled text data. We find that word and phone embeddings can achieve comparable performance without using POS and TOBI features. More promisingly, combining the conventional full feature set with phone embedding, the lowest RMSE is achieved. There is still a substantial amount of work to do in the future. First, further performance gain is expected when acoustic and text features are integrated as input. Second, our work on text-to-articulatory prediction is still preliminary. We plan to take duration and phone state information into account. From [8], a dramatic RMSE decrease is observed if these features are included. Our word/phone vector features need to be finely tuned with different training data and dimensionality. We believe fine-tuning will further boost the performance.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No. 61175018) and the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University.

7. References

[1] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus." in *Proc. INTERSPEECH*. Florence, Italy: ISCA, August 2011, pp. 1505–1508.

[2] J. Sun and L. Deng, "An overlapping-feature-based phonological model incorporating linguistic constraints: Applications to speech recognition," *The Journal of the Acoustical Society of America*, vol. 111, no. 2, pp. 1086–1101, 2002.

[3] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[4] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Integrating articulatory features into hmm-based parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.

[5] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[6] A. Ben-Youssef, H. Shimodaira, and D. A. Braude, "Speech driven talking head from estimated articulatory features," in *Proc. ICASSP*. Florence, Italy: IEEE, May 2014, pp. 4573–4577.

[7] K. Zhao, Z.-Y. Wu, and L.-H. Cai, "A real-time speech driven talking avatar based on deep neural network," in *Proc. APSIPA*. Kaohsiung, Taiwan: IEEE, October 2013, pp. 1–4.

[8] Z.-H. Ling, K. Richmond, and J. Yamagishi, "An analysis of hmm-based prediction of articulatory movements," *Speech Communication*, vol. 52, no. 10, pp. 834–846, 2010.

[9] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory hmm," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[10] K. Richmond, "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion," in *Advances in Nonlinear Speech Processing*. Springer, 2007, pp. 263–272.

[11] K. Richmond, Z.-H. Ling, J. Yamagishi, and B. Uria, "Preliminary inversion mapping results with a new ema corpus," in *Proc. INTERSPEECH*. Brighton, UK: ISCA, September 2009, pp. 2835C–2838.

[12] S. T. Roweis, "Data driven production models for speech processing," Ph.D. dissertation, California Institute of Technology, 1999.

[13] S. Hiroya and M. Honda, "Estimation of articulatory movements from speech acoustics using an hmm-based speech production model," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 175–185, 2004.

[14] K. Richmond, Z.-H. Ling, J. Yamagishi, and B. Uria, "On the evaluation of inversion mapping performance in the acoustic domain," in *Proc. INTERSPEECH*. Lyon, France: ISCA, August 2013, pp. 1012–1016.

[15] G. Papcun, J. Hochberg, T. R. Thomas, F. Laroche, J. Zacks, and S. Levy, "Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data," *The Journal of the Acoustical Society of America*, vol. 92, no. 2, pp. 688–700, 1992.

[16] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, University of Edinburgh, 2002.

[17] K. Richmond, S. King, and P. Taylor, "Modelling the uncertainty in recovering articulation from acoustics," *Computer Speech & Language*, vol. 17, no. 2, pp. 153–172, 2003.

[18] K. Shirai and T. Kobayashi, "Estimating articulatory motion from speech wave," *Speech Communication*, vol. 5, no. 2, pp. 159–170, 1986.

[19] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[20] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proc. ICASSP*. Vancouver, BC, Canada: IEEE, May 2013, pp. 8599–8603.

[21] Y. Qian, Y.-C. Fan, W.-P. Hu, and F. K. Soong, "On the training aspects of deep neural network (dnn) for parametric tts synthesis," in *Proc. ICASSP*. Florence, Italy: IEEE, May 2014, pp. 3829–3833.

[22] H. Zen, "Deep learning in speech synthesis," *Proc. ISCA SSW8*, August 2013.

[23] B. Uria, I. Murray, S. Renals, and K. Richmond, "Deep architectures for articulatory inversion," in *Proc. INTERSPEECH*. Portland, Oregon, USA: ISCA, September 2012, pp. 867C–870.

[24] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[25] K. E. 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*, vol. 2. Banff, Alberta, Canada: ISCA, October 1992, pp. 867–870.

[26] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[27] G. E. Hinton, "Distributed representations," 1984.

[28] P.-L. Wang, Y. Qian, and F. K. Soong, "Word embedding for recurrent neural network based tts synthesis," in *ICASSP*. Brisbane,Australia: IEEE, April 2015.

[29] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings," *arXiv preprint arXiv:1502.03520*, 2015.

[30] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[31] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[32] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[33] B. Fan, L.-J. Wang, F. K. Soong, and L. Xie, "Photo-real talking head with deep bidirectional lstm," in *ICASSP*. Brisbane,Australia: IEEE, April 2015.

[34] J. Vaissière, "Language-independent prosodic features," in *Prosody: Models and measurements*. Springer Berlin Heidelberg, 1983, pp. 53–66.

[35] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

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

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

[38] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.