VOWEL DURATION MEASUREMENT USING DEEP NEURAL NETWORKS

Yossi Adi, Joseph Keshet *Matthew Goldrick*

Dept. of Computer Science
Bar-Ilan University, Ramat-Gan, Israel
adiyoss@cs.biu.ac.il, joseph.keshet@biu.ac.il

Dept. of Linguistics
Northwestern University, Evanston, IL, USA
matt-goldrick@northwestern.edu

ABSTRACT

Vowel durations are most often utilized in studies addressing specific issues in phonetics. Thus far this has been hampered by a reliance on subjective, labor-intensive manual annotation. Our goal is to build an algorithm for automatic accurate measurement of vowel duration, where the input to the algorithm is a speech segment contains one vowel preceded and followed by consonants (CVC). Our algorithm is based on a deep neural network trained at the frame level on manually annotated data from a phonetic study. Specifically, we try two deep-network architectures: convolutional neural network (CNN), and deep belief network (DBN), and compare their accuracy to an HMM-based forced aligner. Results suggest that CNN is better than DBN, and both CNN and HMM-based forced aligner are comparable in their results, but neither of them yielded the same predictions as models fit to manually annotated data.

Index Terms— vowel duration measurement, convolution neural networks, deep belief networks, hidden Markov models, forced alignment

1. INTRODUCTION

Vowel durations are often measured in studies addressing specific issues in phonetics [1, 2, 3]. These typically utilize vowel durations as a dependent measure, examining how duration changes across situations, e.g., when vowels are elicited in different contexts, produced by different speakers, etc.

To obtain accurate data most researchers have relied on manual annotation. This approach is clearly not ideal: it is highly resource intensive and fundamentally subjective. To address these issues, recent phonetic studies have used computational methods to measure acoustic properties of speech automatically, e.g., [2, 4, 5]. These methods greatly reduce the resources required as well as minimizing the role of subjective judgments.

In this paper we try to address the problem of automatic measurement of vowel duration. Most of the work related to vowel duration measurement was done using HMM-based forced alignment. However, those aligners require the phonetic transcription of the input signal, and should be carefully tuned [6]. Our work is focused on input of speech segments where a single vowel is preceded and followed by consonant (CVC), and the phonetic transcription is not needed.

Here we take a different route and train a classifier at the frame-level to detect whether the frame is a vowel or not. We used state-of-the-art deep neural network (DNN) as a classifier, comparing two DNN architectures: deep belief network (DBN) and convolutional neural network (CNN). Both architectures have produced good results in previous speech processing studies [7, 8]. Each architecture was trained on manually annotated data and their performance was compared. At inference time, the classifier predicts the probability of each frame of the input as being a vowel. The predictions are smoothed out to have a single chunk representing the vowel, and then vowel duration is computed.

We compare the accuracy of DBN, CNN and HMM-based force aligner on manually annotated data. The results show that CNN is better than DBN, and the CNN and HMM-based forced aligner are comparable. We further evaluated the performance of CNN and HMM on a phonetic study, which examines how placing words in a context that strongly emphasizes processing of sentence structure would influence speech articulation [3]. The results suggest that the CNN-based classifier is comparable to the HMM-based forced aligner in terms of deviation from the manual annotations. However, it seems that neither method is clearly superior in terms of matching the manual annotations.

The paper is organized as follows. In Section 2 we state the problem definition formally. In Section 3 we present the acoustic feature set and the architecture of the network used. In Section 4 we briefly describe the benchmark data and how it was recorded. We present a detailed experimental results and analysis in Section 5. We conclude the paper in Section 6.

2. PROBLEM SETTING

In the problem of *vowel duration measurement* we are provided with a speech signal which includes exactly one vowel preceded and followed by consonants, often denoted CVC. Our goal is to predict the vowel duration accurately. We de-

978-1-4673-7454-5/15/$31.00 ©2015 IEEE

note the domain of the acoustic feature vectors by $\mathcal{X} \subset \mathbb{R}^d$. The acoustic feature representation of a speech signal is therefore a sequence of vectors $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ where $x_i \in \mathcal{X}$ for all $1 \leq i \leq T$. The length of the input signal varies from one signal to another, thus T is not fixed. We denote by $\mathcal{X}^*$ the set of all finite length sequences over $\mathcal{X}$. In addition, we denote by $t_b \in \mathcal{T}$ and $t_e \in \mathcal{T}$ the vowel onset and offset times, respectively, where $\mathcal{T} = \{1, ..., T\}$. For brevity we set $\boldsymbol{t} = (t_b, t_e)$.

Our goal is to learn a function, denoted f, which takes as input a speech signal $\mathbf{x}$ and returns the pair $\boldsymbol{t}$. The vowel duration can be computed from the output of this function. In other words, $f : \mathcal{X}^* \rightarrow \mathcal{T}^2$ is a function from the domain of all possible CVC speech segments to the domain of all possible onset and offset pairs.

In order to qualify the quality of the prediction we need to define a *measure of performance* or *evaluation metric* between the predicted and the target onset and offset pairs. We denote by $\gamma(\boldsymbol{t}, \hat{\boldsymbol{t}})$ the cost of predicting the pair $\hat{\boldsymbol{t}}$ while the target pair is $\boldsymbol{t}$. Formally, $\gamma : \mathcal{T}^2 \times \mathcal{T}^2 \rightarrow \mathbb{R}$ is a function that gets as input two ordered pairs, and returns a scalar. We assume that $\gamma(\hat{\boldsymbol{t}}, \boldsymbol{t}) \geq 0$ for any two pairs of time sequences, and $\gamma(\boldsymbol{t}, \boldsymbol{t}) = 0$. The cost function we use is:

$$\gamma(\hat{\boldsymbol{t}}, \boldsymbol{t}) = \big[|\hat{t}_b - t_b| - \epsilon_b\big]_+ + \big[|\hat{t}_e - t_e| - \epsilon_e\big]_+, \qquad (1)$$

where $[\pi]_+ = \max\{0, \pi\}$, and ϵ_b, ϵ_e are pre-defined constants. The above function measures the absolute differences between the predicted and the manually annotated vowel onsets and offsets. Since the manual annotations are not exact, we allow a mistake of ϵ_b and ϵ_e frames at the vowel onset and offset respectively, and only panelize predictions that varies by more than ϵ_b or ϵ_e frames.

Our training algorithm is based on training set of m examples, $S = \{(\mathbf{x}_1, \boldsymbol{t}_1), \ldots, (\mathbf{x}_m, \boldsymbol{t}_m)\}$, which were manually labeled. See Section 4 for a detailed description of the data. Ideally, we would like to evaluate our predictions against the manually annotated predictions using a cost function that does take into account small variations in annotations. In the next section we show how we find a function f from this set.

3. THE ARCHITECTURE

In this section we describe the two network architectures that can be used as frame-level classifier for vowel detection. We start by presenting the acoustic features we used as the input for both architectures.

We extracted standard MFCC features (with energy, delta and delta-delta), and concatenate to each frame the previous and the next two frames. We also added the normalized pitch [9] as an additional feature. The features were extracted in frames of 10 msec and spanned a window of 25 msec. Overall we had d = 196 features (5×39 MFCC + 1 pitch).

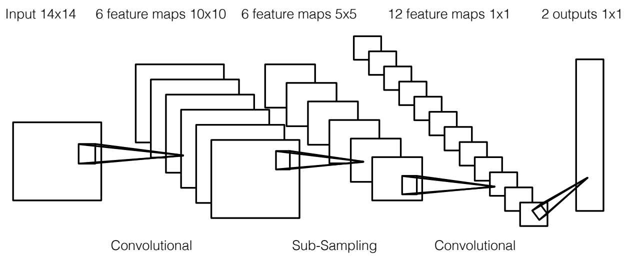

Fig. 1. *CNN architecture*

3.1. Convolutional neural network

Our first architecture is based on a variation of LeNet1 CNN [7]. The input of the network is a matrix $\mathbf{x} \in \mathbb{R}^{14\times14}$ (we reshape the input 196 features into a 14×14 matrix), and the output is a number $y \in [0, 1]$, indicating the probability that a given speech frame is a vowel. Our network is composed from four learned layers: two convolutional layers, a subsampling layer and an output layer.

The first layer is a convolutional layer which has 6 feature maps. Those are connected to the input layer using 6 kernels of size 5×5. The second layer is a sub-sampling layer, we use a 2×2 mean-polling-layer. The third layer has 12 feature maps which are fully connected to all 6 mean-pooling-layer using 72 kernels of size 5×5. Finally, we feed the output layer with the output of the third layer. The output layer consists two neurons corresponding to the occurrence of the vowel on this frame. A simplified description of the networks can be viewed in Figure 1.

The network is trained so as to minimize the mean square error using stochastic gradient descent with mini-batches of size m_b, namely,

$$\frac{1}{m_b} \sum_{i=1}^{m_b} (y_i - \hat{y}_i)^2, \qquad (2)$$

where $y_i \in \{0, 1\}$ is 1 if the frame i is annotated as a vowel and 0 otherwise, and $\hat{y}_i \in (0, 1)$ is the network prediction. The network was trained on 36,000 frames, a mini-batch size of m_b=50, fixed learning rate equal to 1 and 100 epochs. Our implementation is based on a modified versions of the code in [10]. All parameters were chosen on a validation set.

3.2. Deep belief network

Our second architecture is based on DBN. The network is composed of five layers: an input layer, 3 hidden layers, and an output layer. The input layer is composed of 196 inputs. The first and second hidden layers are composed of 500 hidden units, the third hidden layer is composed of 2000 hidden units and the output layer was uses softmax activation function. A simplified description of the networks can be viewed in Figure 2.

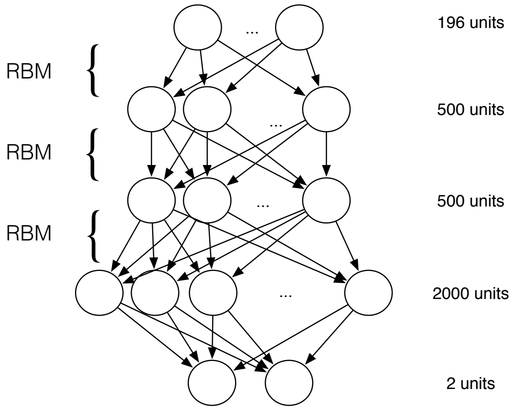

Fig. 2. *DBN architecture*

Due to the highly non-linear functions involved, DNNs are difficult to train directly by stochastic gradient descent. Hence, each layer is pre-trained in an unsupervised way to model the previous layer expectation. In this work, we use restricted Boltzmann machines (RBM) [11] to model the joint distribution of the previous layers units in a DBN [12]. The network is trained so as to minimize the cross entropy using stochastic gradient descent with a line search:

$$-\sum_{i=1}^{m_b} y_i \log \hat{y}_i, \tag{3}$$

where y_i and $\hat{y}_i$ are the i-th frame annotation and the network prediction, respectively. We pre-trained the system with 50 epochs per each layer, and 100 epochs for fine-tuning feed-forward training. The network was trained on 36,000 frames, with a mini-batch size of m_b=100, and a learning rate was 0.1. Our implementation is based on the code in published by R. Salakhutdinov and G. Hinton[1]. All parameters were chosen on a validation set.

3.3. Smoothing

We converted the network output from a vector of probabilities into a pair of vowel onset and offset. We rounded the probability of each frame to be either zero or one. Then we smoothed the resulted sequence to have a single chunk, representing a continues vowel.

Notice that the function each network is trained to minimize is different then in the desired cost function in (1). The difference is due to the fact the the network examine single frames at a time, and not build to work with variable length inputs. Nevertheless, it turns out that while minimizing (2) or (3) we get also good results when measuring the performance using (1).

4. DATASET

The dataset used in our experiments contains speech segments that were composed from 64 native English speakers (55 female) aged 18-34 with no history of speech or language deficits. To obtain the recordings, participants were asked to name aloud the noun depicted by a picture in two different ways: with the picture present alone and when the picture occurred at the end of a sentence. They were instructed to produce the name as quickly as possible. To avoid misunderstanding the participants were first familiarized with the pictures. Each recorded segment contains one English CVC noun, where the list of vowels are: /i, ɛ, ae, ɑ, o, u/. The total size of the data set is 2684 CVC speech segments.

Vowel duration was hand-measured in Praat [13]. Vowel onsets and offsets were marked using cues from the waveform and spectrogram [14]. A second coder marked 25 % of the data to assess measurement reliability. Measurements were well correlated, $r(627) = .84$, $p < .0001$. A detailed description of the dataset can be found in [3].

5. EXPERIMENTAL RESULTS

We compare the performance of both DNN architectures and an HMM-based forced aligner to manual benchmark data from picture naming of English monosyllabic words describe in the previous section.

Forced alignment is an algorithm to align a sequence of phonemes (or words) to the speech signal. The force alignment gets as input a speech signal and a sequence of phonemes (or words) uttered in the signal. The output of the aligner is the location of each phoneme in the speech. The forced alignment is based on an HMM trained as a phoneme recognizer. In our experiments we used Penn Aligner [15], which is HMM-based phoneme aligner trained with the HTK toolkit[2].

5.1. Baseline

First we compare the results of CNN and DBN to HMM on the cost defined in (1). The result are presented in Table 1. Each row in the table, corresponds to different values of ϵ_b and ϵ_e, and the results of the deviations in msec of the onset and the offset. For comparison, the inter-transcriber deviations are on average 3.5 msec for the onset and 20 msec for the offset (ϵ_b and ϵ_e are both 0).

It can be seen from the table that the performance of HMM is slightly better than the performance of CNN except for the onset in the higher values of ϵ_b and ϵ_e and much better then the results of the DBN. Nevertheless, the CNN and DBN does not need as input the sequence of uttered phonemes, while the HMM does.

[1] `http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html`

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

Table 1. *Results of CNN, DBN and HMM relative to manual annotation. Average deviation of onset and offset for different values of ϵ_b and ϵ_e [in msec].*

		CNN		DBN		HMM	
ϵ_b	ϵ_e	onset	offset	onset	offset	onset	offset
0	0	21	36	62	106	15	25
10	15	20	33	61	102	12	20
20	40	7.3	22	60	98	9	12
30	50	3.5	19	55	97	6	10

Since both network architectures are not aimed to minimize (1), we give here the misclassification error rate as well. The CNN reaches misclassification rate of 4.57% while the DBN reaches 22%, both of these results were measured on the test set.

In the next subsections we perform a deeper analysis on the results of CNN comparing them manually labeled ones. We choose to preform the analysis on the CNN predictions due to the poor results of the DBN.

5.2. Measurement Deviation

Figure 3 provides a comparison of the vowel durations from the manual annotators (y-axis) vs. each algorithm (x-axis).

The mean squared error relative to the manual benchmark was slightly higher for CNN (2.4 msec^2) vs. HMM (1.8 msec^2). To assess the reliability of this difference, the distribution of differences was estimated using 1,000 bootstrap samples (created by resampling the observed differences in mean squared error across the methods). The mean difference of 0.6 msec^2 was reliable (95% confidence interval: [0.3, 0.9]).

5.3. Model-Based Comparison

As noted in the introduction, vowel durations are used in studies addressing specific phonetic issues. These typically utilize vowel durations as a dependent measure, examining how duration is modulated by properties of: the context in which vowels appear; the individuals producing the vowels; and the particular stimuli that were used to elicit the productions. Statistical models are used to assess the importance of these factors. To illustrate, the study that provides the benchmark data used here [3] examined how placing words in a context that strongly emphasizes processing of sentence structure would influence speech articulation. Speakers named a set of pictures both in isolation (where there is no need to process sentence structure) as well as following a sentence frame (strongly emphasizing the processing of sentence structure). In the speech field, hierarchical mixed-effects regressions [16] are the current state-of-the-art analytic technique for assessing the reliability of such effects. These allow estimation of the effect of the variable of interest (e.g., context of naming) while controlling for other properties. In this case, the analysis [17] revealed that context had a reliable effect (such that vowel durations were shorter when the picture was named following a sentence context vs. in isolation).

Given the critical role that such statistical models play in utilizing vowel duration data, our second evaluation method to compares the properties of statistical models fit to manually annotated data vs. data obtained from CNN or HMM.

5.3.1. Regression Model Structure

The source study here [3] manipulated two factors: the production context (picture naming in isolation vs. following a sentence) and the number of words phonologically similar to the target that share its grammatical category (*lexical density*). To account for these factors, the model included a centered density measure which interacted with a contrast-coded fixed effect reflecting production context (isolation vs. sentence). Additional contrast-coded factors controlled for block in which the picture was presented (first vs. second), as well as the target vowel identity.

To control for idiosyncratic contributions from the random sample of speakers (e.g., the specific individuals tested here vs. all English speakers) and stimuli (e.g., the specific words used here vs. all words of English), the model also included crossed random effects. These included random intercepts for participants and words, along with uncorrelated random slopes for context and density by participant and context by word.

To control for skew in the dependent measure, vowel durations were log transformed prior to analysis. To control for outliers, all models were refit after excluding observations with standardized residuals greater than 2.5 [18]. To assess whether a given factor made a significant contribution to variance in vowel duration, the likelihood ratio test was used to compare models with vs. without the factor [19].

5.3.2. Comparison of Model Parameters

Table 2 provides the parameter estimates for fixed effects. Several features of the manual annotation results are reflected in models fit by both annotation methods. Most parameters have the same sign. All methods show a significant shortening for naming in sentence contexts vs. isolation ($\chi^2(1)s > 7.4, ps < .01$).

There are also significant divergences. While all models show that durations are longer in the second block, only CNN found a (marginally) significant effect ($\chi^2(1)s > 3.81, ps < .051$). Similarly, while all models show shorter durations for vowels in words with more lexical neighbors, only the HMM found a significant effect ($\chi^2(1)s > 5.63, ps < .02$). The HMM also found significant effects of two factors related to vowel identity ($\chi^2(1)s > 4.15, ps < .05$). Thus, by this

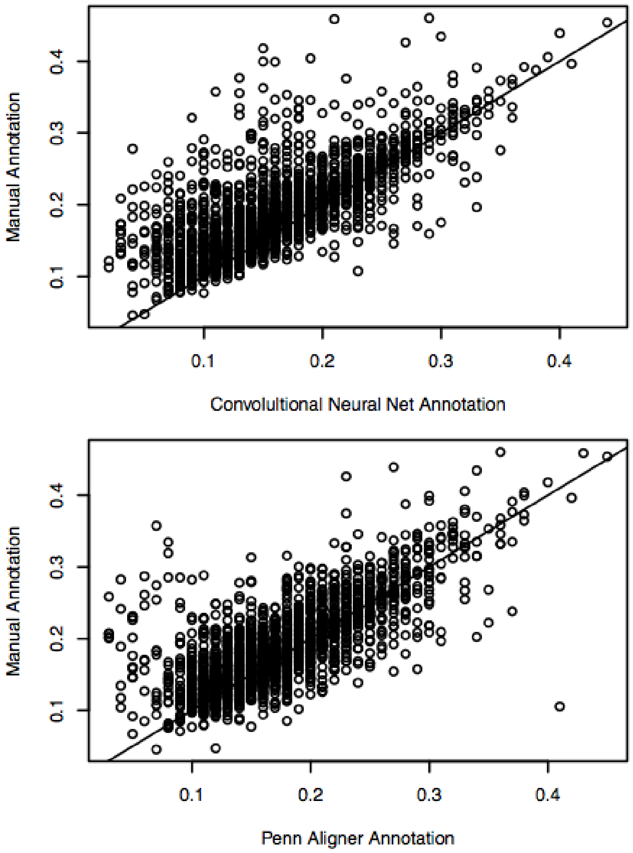

Fig. 3. *Scatterplot of algorithm vs. manual annotation vowel durations (in seconds)*

method of drawing inferences from statistical models, CNN lead to one false positive effect while HMM leads to three false positives (out of 10 possible effects).

5.3.3. Comparison of Model Predictions

While phonetic studies typically draw inferences based on assessments of individual predictors, an alternative means of assessing such models is examining their predictions on novel data. To generate such predictions, we performed 4-fold cross-validation. We maintained a roughly even balance of observations within each participant across high vs. low density items (using a median split) and production contexts.

Figure 4 provides a comparison of model predictions. The mean squared error, relative to the manual model, showed that models fit to the automated data showed substantial deviations in predictions. The model fit to CNN annotations showed a mean squared error of 334 log msec^2, significantly greater than that of the HMM model (65 log msec^2; bootstrapped 95% CI for difference [237, 302]). While both models show divergence from the manual model, the HMM model's predictions are more similar to than those of the CNN model.

Table 2. *Table 1. Estimates for the effect of each fixed parameter on log vowel duration (in seconds).* ***Bolded*** *parameters make significant contributions to variance.*

		Estimate		
	Parameter	Manual	CNN	HMM
	Intercept	-1.7046	-1.861	-1.7768
	Production Context	**-0.0561**	**-0.1414**	**-0.071**
	Lexical Density	-0.0117	-0.0075	**-0.0151**
	Density × Context	0.0002	-0.0001	-0.004
	Block	0.0109	**0.0379**	0.0096
Vowel	ɑ vs. ae	0.0664	0.1232	0.035
	ɛ vs. ɑ, ae	-0.0538	-0.0307	**-0.0584**
	i vs. ɛ,ɑ,ae	-0.0393	-0.0429	**-0.0346**
	o vs. i,ɛ,ɑ,ae	0.0025	-0.0117	0.0091
	u vs. all others	0.0005	-0.0052	-0.022

6. DISCUSSION AND FUTURE WORK

While both HMM and CNN annotations yield duration values that are, on average, quite similar to manual annotations, neither method yields the same conclusions as would be drawn from manually-annotated data. In our test set, both algorithms recovered an effect that was reliable in manually-annotated data; however, both also yielded false positives. Moreover, both automatic annotation methods yielded different predictions than models fit to manually annotated data. Neither method was clearly superior in terms of matching the manual annotations; CNN resulted in fewer incorrect inferences regarding significant effects, but the HMM model predictions were clearly more like those of models fit to manually annotated data.

We believe these results are promising since they show that neural networks, especially CNNs, can reach performance comparable to HMM-based models with no need for phonetic transcription. However, we would emphasize that these are initial results for the CNN and there is much room for exploring new architectures in the neural network field such as deeper networks with more layers or hidden units. The most promising direction would be focused on recurrent neural networks.

Acknowledgements: Funded in part by NIH-NICHD grant HD077140.

7. REFERENCES

[1] Benjamin Munson and Nancy Pearl Solomon, "The effect of phonological neighborhood density on vowel articulation," *Journal of speech, language, and hearing research*, vol. 47, no. 5, pp. 1048–1058, 2004.

[2] Susanne Gahl, Yao Yao, and Keith Johnson, "Why reduce? phonological neighborhood density and phonetic

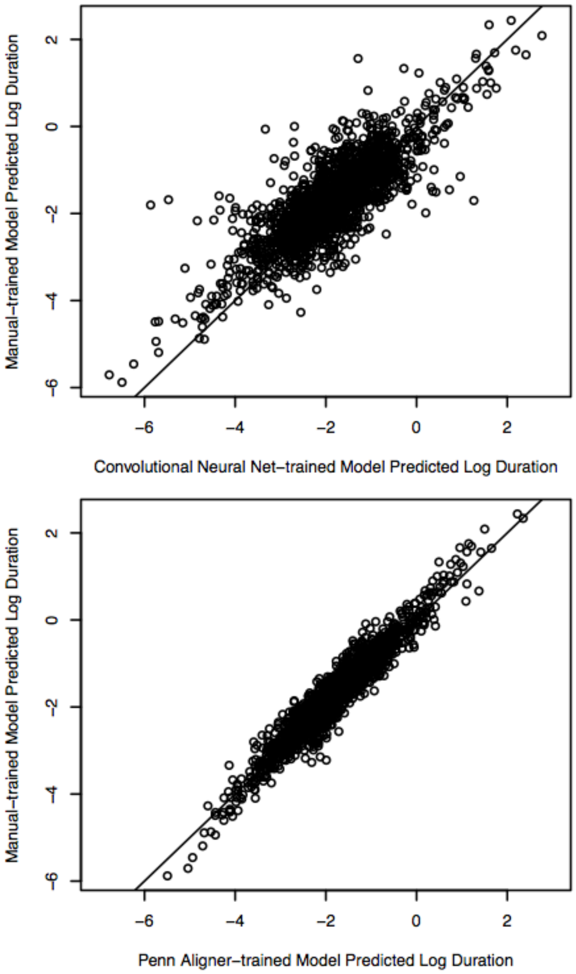

Fig. 4. *Scatterplot of predicted vowel durations (in log seconds) from regression models fit to algorithm vs. manual annotation*

reduction in spontaneous speech," *Journal of Memory and Language*, vol. 66, no. 4, pp. 789–806, 2012.

[3] Jordana R Heller and Matthew Goldrick, "Grammatical constraints on phonological encoding in speech production," *Psychonomic bulletin & review*, vol. 21, no. 6, pp. 1576–1582, 2014.

[4] William Labov, Ingrid Rosenfelder, and Josef Fruehwald, "One hundred years of sound change in philadelphia: Linear incrementation, reversal, and reanalysis," *Language*, vol. 89, no. 1, pp. 30–65, 2013.

[5] Jiahong Yuan and Mark Liberman, "F0 declination in english and mandarin broadcast news speech.," in *INTERSPEECH*. Citeseer, 2010, pp. 134–137.

[6] Keelan Evanini, *The permeability of dialect boundaries: A case study of the region surrounding Erie*, Ph.D. thesis, University of Pennsylvania, 2009.

[7] Yann LeCun, LD Jackel, L Bottou, A Brunot, C Cortes, JS Denker, H Drucker, I Guyon, UA Muller, E Sackinger, et al., "Comparison of learning algorithms for handwritten digit recognition," in *International conference on artificial neural networks*, 1995, vol. 60, pp. 53–60.

[8] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara 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.

[9] David Talkin, "A robust algorithm for pitch tracking," *Speech coding and synthesis*, vol. 495, pp. 518, 1995.

[10] Rasmus Berg Palm, "Prediction as a candidate for learning deep hierarchical models of data," M.S. thesis, Technical University of Denmark, 2012.

[11] Paul Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," 1986.

[12] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural comp.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] Paul Boersma and David Weenink, "Praat, a system for doing phonetics by computer," 2001.

[14] Gordon E Peterson and Ilse Lehiste, "Duration of syllable nuclei in english," *The Journal of the Acoustical Society of America*, vol. 32, no. 6, pp. 693–703, 1960.

[15] Jiahong Yuan and Mark Liberman, "Speaker identification on the scotus corpus," *Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3878, 2008.

[16] R Harald Baayen, Douglas J Davidson, and Douglas M Bates, "Mixed-effects modeling with crossed random effects for subjects and items," *Journal of memory and language*, vol. 59, no. 4, pp. 390–412, 2008.

[17] Jordana R Heller and Matthew Goldrick, "Corrigendum to 'grammatical constraints on phonological encoding in speech production'," *Psychonomic bulletin & review*, 2014.

[18] Rolf Harald Baayen, *Analyzing linguistic data: A practical introduction to statistics using R*, Cambridge University Press, 2008.

[19] Dale J Barr, Roger Levy, Christoph Scheepers, and Harry J Tily, "Random effects structure for confirmatory hypothesis testing: Keep it maximal," *Journal of memory and language*, vol. 68, no. 3, pp. 255–278, 2013.