Automatic Measurement of Positive and Negative Voice Onset Time

Katharine Henry[1], *Morgan Sonderegger*[1] *and Joseph Keshet*[2]

[1]University of Chicago, Chicago, Illinois, USA
[2]TTI-Chicago, Chicago, Illinois, USA

kehenry@uchicago.edu, morgan@cs.uchicago.edu, jkeshet@ttic.edu

Abstract

Previous work on automatic VOT measurement has focused on positive-valued VOT. However, in many languages VOT can be either positive or negative ("prevoiced"). We present a discriminative algorithm that simultaneously decides whether a stop is prevoiced and measures its VOT. The algorithm operates on feature functions designed to locate the burst and voicing onsets in the positive and negative VOT cases. Tested on a database of positive- and negative-VOT voiced stops, the algorithm predicts prevoicing with $>$90% accuracy, and gives good agreement between automatic and manual measurements.

Index Terms: voice onset time, automatic phonetic measurement, discriminative methods, structured prediction

1. Introduction

Voice onset time (VOT), the time between the onset of a stop burst and the onset of voicing, is an important cue to stop voicing and place. It is widely measured in theoretical and clinical settings, for example to characterize how communication disorders affect speech [1] or how languages differ in the phonetic cues to stop contrasts [2, 3]; it is also increasingly used as a feature for ASR tasks such as stop consonant classification [4, 5, 6]. Automatic VOT measurement would be very beneficial for clinical and theoretical studies, where it is currently usually measured manually, and is essential for ASR applications.

Several recent studies have proposed VOT measurement algorithms [5, 6, 7, 8],[1] all making the assumption that VOT is positive (burst onset precedes voicing onset). However, VOT can in general also be negative (voicing onset precedes burst onset), in which case the stop is "prevoiced." In English, for example, voiceless stops (/p/, /t/, /k/) always have positive VOT, while voiced stops (/b/, /d/, /g/) can have positive or negative VOT. In Dutch and French, voiced stops usually have negative VOT, while voiceless stops have positive VOT [9]. The spectral cues that indicate negative VOT differ considerably from those used to identify positive VOT, for example due to the presence of more low-frequency energy when detecting a voiced rather than voiceless burst. To handle an arbitrary stop consonant, a VOT measurement algorithm must perform two tasks: decide whether the VOT is positive or negative and return a VOT measurement.

We present an extension of the algorithm for positive VOT-only in [7] that performs these tasks, and can therefore be applied to both voiced and voiceless stops. Given a set of labeled training data containing both positive and negative VOT examples, two classifiers are learned. Applied to a speech segment containing a stop burst, the classifiers determine the most likely positive and negative VOT values, as well as a confidence measure for each. The stop's VOT is predicted to be the value with the higher confidence. The classifiers operate on two sets of customized features based on spectro-temporal cues to the location of the burst and voicing onsets in the positive and negative VOT cases.

2. Problem definition

The input to our algorithm is a speech segment containing a single stop consonant, and the output is the sign and absolute value of the stop's VOT. The speech segment can be of arbitrary length, and its beginning need not be synchronized with the burst onset, the voicing onset, or the closure; it is only required that the segment begins before and ends after the burst onset and the voicing onset.

Let $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ denote the speech segment, represented as a sequence of acoustic feature vectors, where each $\mathbf{x}_t \in \mathbb{R}^D$ $(1 \leq t \leq T)$ is a D-dimensional vector. The length of the speech segment, T, is not a fixed value since speech segments can have different durations. Each speech segment is associated with a pair of numbers: the onset of the burst, $t_b \in \mathcal{T}$, and the onset of the voicing, $t_v \in \mathcal{T}$, where $\mathcal{T} = \{1, \ldots, T\}$. We call this pair an *onset pair*. We distinguish between two types of stop realization: the vocal cords may begin vibrating after the burst onset ($t_b < t_v$: positive VOT), or voicing may begin before the burst onset ($t_b > t_v$: negative VOT). We denote by $s \in \mathcal{S}$ the type of stop realization, where $\mathcal{S} = \{pos, neg\}$.

Our goal is to learn a function $f : \mathcal{X}^* \to \mathcal{T} \times \mathcal{T} \times \mathcal{S}$, that maps the domain of all speech segments to the domain of all onset pairs and stop realizations. Given in-

[1] This list is not exhaustive, due to space considerations.

put speech segment, $\bar{\mathbf{x}}$, we define the cost of predicting $(\hat{t}_b, \hat{t}_v, \hat{s}) = f(\bar{\mathbf{x}})$, when the manual annotation is (t_b, t_v, s), using the following cost function

$$\gamma\left((t_b, t_v, s), (\hat{t}_b, \hat{t}_v.\hat{s})\right) = \\ \begin{cases} \max\{|(\hat{t}_v - \hat{t}_b) - (t_v - t_b)| - \gamma_0, 0\} & s = \hat{s} \\ \gamma_m & s \neq \hat{s} \end{cases} \quad (1)$$

γ_m is a user-defined parameter that penalizes an incorrect prediction of the VOT's sign. The threshold γ_0 is a user-defined parameter that allows small annotation inaccuracies and inconsistencies, given that the VOT's sign is correctly predicted. If we set γ_0 to 3 msec, for example, then differences between the manually labeled VOT $(t_v - t_b)$ and predicted VOT $(\hat{t}_v - \hat{t}_b)$, which are less than 3 msec, are not counted in the cost function. The goal of the learning algorithm is to find the function f that minimizes the expected cost, where the expectation is taken with respect to a fixed but unknown distribution over speech segments and the onset pairs. In the next section we present a learning algorithm that aims to minimize the expected cost.

3. Discriminative learning

Similarly to previous work in structured prediction [10, 11], the function f is constructed from a predefined set of N feature maps $\{\phi_i\}_{i=1}^N$, each of the form $\phi_i : \mathcal{X} \times \mathcal{T} \times \mathcal{T} \times \mathcal{S} \to \mathbb{R}^N$, and a weight vector $\mathbf{w} \in \mathbb{R}^N$. The function is a linear decoder of the following form

$$(\hat{t}_b, \hat{t}_v, \hat{s}) = \arg \max_{(t_b, t_v, s)} \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, t_b, t_v, s), \quad (2)$$

where we have used vector notation for the feature maps $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_N)^\top$. This vector-valued function is used to map the variable length speech segment along with a onset pair and stop realization type to an abstract vector space in $\mathbb{R}^N$. For a given speech segment, $\bar{\mathbf{x}}$, each onset pair (t_b, t_v) and realization type s correspond to a single vector in $\mathbb{R}^N$. The algorithm presented here should set the weights $\mathbf{w}$ such that the projection of $\mathbf{w}$ onto the vector corresponding to the manually labeled onset pair (t_b, t_v) and correct realization type s should be maximized relative to the vectors corresponding to all other onset pairs and realization types.

3.1. Iterative algorithm

Recall that our learning algorithm receives as input a training set $S = \{(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}, s_i)\}_{i=1}^m$ and returns a weight vector $\mathbf{w}$, which defines the decoding function in Eq. (2). The weight vector is learned using an iterative algorithm based on the family of algorithms described in [12] for structured prediction. Let $\mathbf{w}_t$ be the weight vector after the t^{th} iteration, and let $\mathbf{w}_0 = \mathbf{0}$. On each iteration we consider a single example $(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}, s_i)$, and use the current weight vector $\mathbf{w}_t$ to predict its VOT and realization type $\bar{\mathbf{x}}_i$ as:

$$(\hat{t}_b, \hat{t}_v, \hat{s}) = \arg \max_{(t_b, t_v, s)} \mathbf{w}_t \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i, t_b, t_v, s) \\ + \gamma\left((t_{b_i}, t_{v_i}, s_i), (\hat{t}_b, \hat{t}_v, \hat{s})\right). \quad (3)$$

Next, we update the weight vector $\mathbf{w}_{t+1}$ to be

$$\mathbf{w}_{t+1} \ = \ \mathbf{w}_t + \tau_t \, \Delta\boldsymbol{\phi}_t \ , \quad (4)$$

where $\Delta\boldsymbol{\phi}_t = \boldsymbol{\phi}(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}, s_i) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i, \hat{t}_b, \hat{t}_v, \hat{s})$, and

$$\tau_t = \frac{\gamma\left((t_{b_i}, t_{v_i}, s_i), (\hat{t}_b, \hat{t}_v, \hat{s})\right) - \mathbf{w}_t \cdot \Delta\boldsymbol{\phi}_t}{\|\Delta\boldsymbol{\phi}_t\|^2}.$$

In words, we add to $\mathbf{w}_t$ a vector which is a scaled version of the difference between the feature maps resulting from the manual annotation $\boldsymbol{\phi}(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}, s_i)$ and from the prediction $\boldsymbol{\phi}(\bar{\mathbf{x}}_i, \hat{t}_b, \hat{t}_v, \hat{s})$.

3.2. Decomposition by stop realization type

The feature maps take as input a speech segment $\bar{\mathbf{x}}$, an onset pair (t_b, t_v) and a stop realization type s. They are designed to have high values when (t_b, t_v) and s make sense given $\bar{\mathbf{x}}$ and to have lower values otherwise. We decompose the vector-valued function $\boldsymbol{\phi}$ into two portions: $\boldsymbol{\phi}^+ : \mathcal{X} \times \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{N^+}$, which is used when $t_b < t_v$, and $\boldsymbol{\phi}^- : \mathcal{X} \times \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{N^-}$, which is used when $t_v < t_b$, with $N = N^+ + N^-$. That is,

$$\boldsymbol{\phi}(\bar{\mathbf{x}}, t_b, t_v, s) = \begin{bmatrix} \mathbb{1}_{\{s=pos\}} \, \boldsymbol{\phi}^+(\bar{\mathbf{x}}, t_b, t_v) \\ \mathbb{1}_{\{s=neg\}} \, \boldsymbol{\phi}^-(\bar{\mathbf{x}}, t_b, t_v) \end{bmatrix}$$

when $\mathbb{1}_{\{x\}}$ is the indicator function. The weight vector $\mathbf{w}$ can similarly be split into two parts: $\mathbf{w} = (\mathbf{w}^+, \mathbf{w}^-)^\top$.

The prediction of Eq. (2) is now performed in two phases. We first predict the positive and negative VOTs:

$$(t_b^+, t_v^+) \ = \ \arg \max_{(t_b, t_v)} \mathbf{w}^+ \cdot \boldsymbol{\phi}^+(\bar{\mathbf{x}}, t_b, t_v) \quad (5)$$

$$(t_b^-, t_v^-) \ = \ \arg \max_{(t_b, t_v)} \mathbf{w}^- \cdot \boldsymbol{\phi}^-(\bar{\mathbf{x}}, t_b, t_v). \quad (6)$$

We then predict the stop realization type based on the confidences of these predictions. If $\mathbf{w}^+ \cdot \boldsymbol{\phi}^+(\bar{\mathbf{x}}, t_b^+, t_v^+) > \mathbf{w}^- \cdot \boldsymbol{\phi}^-(\bar{\mathbf{x}}, t_b^-, t_v^-)$ then $(\hat{t}_b, \hat{t}_v) = (t_b^+, t_v^+)$ and $s = pos$, otherwise $(\hat{t}_b, \hat{t}_v) = (t_b^-, t_v^-)$ and $s = neg$. Under this decomposition, the update rule of Eq. (4) becomes

$$\mathbf{w}_{t+1}^+ \ = \ \mathbf{w}_t^+ + \tau_t \Delta\boldsymbol{\phi}_t^+ \quad (7)$$

$$\mathbf{w}_{t+1}^- \ = \ \mathbf{w}_t^- + \tau_t \Delta\boldsymbol{\phi}_t^-, \quad (8)$$

where $\Delta\boldsymbol{\phi}_t^+ = \boldsymbol{\phi}^+(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}) - \boldsymbol{\phi}^+(\bar{\mathbf{x}}_i, \hat{t}_b^+, \hat{t}_v^+)$ and $\Delta\boldsymbol{\phi}_t^- = \boldsymbol{\phi}^-(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}) - \boldsymbol{\phi}^-(\bar{\mathbf{x}}_i, \hat{t}_b^-, \hat{t}_v^-)$, and under the convention that $\boldsymbol{\phi}^+(\bar{\mathbf{x}}, t_b, t_v) = \mathbf{0}$ for $t_v \leq t_b$ and $\boldsymbol{\phi}^-(\bar{\mathbf{x}}, t_b, t_v) = \mathbf{0}$ when $t_b \leq t_v$.

Two cases will serve to exemplify the update rule.

Case I: An example where $t_{b_i} < t_{v_i}$, so $s_i = pos$. Assume the classifier $\mathbf{w}_t$ labels the speech segment $\bar{\mathbf{x}}_i$ as $\hat{s} = neg$, hence $\hat{t}_v < \hat{t}_b$. The update rule is then

$$\begin{aligned}\mathbf{w}^+_{t+1} &= \mathbf{w}^+_t + \tau_t \boldsymbol{\phi}^+(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}) \\ \mathbf{w}^-_{t+1} &= \mathbf{w}^-_t - \tau_t \boldsymbol{\phi}^-(\bar{\mathbf{x}}_i, \hat{t_b^-}, \hat{t_v^-}),\end{aligned}$$

where we set $\boldsymbol{\phi}^+(\bar{\mathbf{x}}, \hat{t}_b, \hat{t}_v) = \mathbf{0}$ because $\hat{t}_v < \hat{t}_b$, and $\boldsymbol{\phi}^-(\bar{\mathbf{x}}, t_{b_i}, t_{v_i}) = \mathbf{0}$ because $t_{b_i} < t_{v_i}$. The update rule *increases* the weight vector $\mathbf{w}^+$, since this example's labeled VOT was positive, and *decreases* the weight vector $\mathbf{w}^-$, which was too high relative to $\mathbf{w}^+$ for this example.

Case II: An example where $t_{b_i} < t_{v_i}$ ($s_i = pos$). Assume the classifier labels this example as $\hat{t}_b < \hat{t}_v$ ($\hat{s} = pos$), but the cost in the prediction is not zero: $|(\hat{t}_v - \hat{t}_b) - (t_v - t_b)| > \gamma_0$. The update rule is then

$$\begin{aligned}\mathbf{w}^+_{t+1} &= \mathbf{w}^+_t + \tau_t \left[\boldsymbol{\phi}^+(\bar{\mathbf{x}}_i, t_{b_i}, t_{v_i}) - \boldsymbol{\phi}^+(\bar{\mathbf{x}}_i, \hat{t}_b, \hat{t}_v)\right] \\ \mathbf{w}^-_{t+1} &= \mathbf{w}^-_t \,.\end{aligned}$$

The update rule adjusts the positive weight vector $\mathbf{w}^+$ to predict positive VOT more accurately, and the negative weight vector $\mathbf{w}^-$ is left intact. This case demonstrates a nice property of the algorithm: if trained on only examples of positive (or negative) VOT, it reduces to the algorithm presented in [7].

3.3. Feature maps

Similarly to [7], seven (D=7) features are extracted from the speech signal every 1 ms. The first five features refer to an STFT taken with a 5 ms Hamming window: the total spectral energy (E_{total}), energy between 50–1000 Hz (E_{low}), energy above 3000 Hz (E_{high}), Wiener entropy (H_{wiener}), and the number of zero crossings of the signal (ZC). Features 6–7 are the maximum of the FFT of the autocorrelation function of the signal from 6 ms before to 18 ms after the frame center (R_l), and a binary voicing detector based on the RAPT pitch tracker [13], smoothed with a 5 ms Hamming window (V).

The vector $\boldsymbol{\phi} = (\boldsymbol{\phi}^+, \boldsymbol{\phi}^-)^\top$ contains N=112 feature maps, 59 of which ($\boldsymbol{\phi}^+$) are used to estimate t_b and t_v for the positive VOT case, and 54 of which ($\boldsymbol{\phi}^-$) are used to estimate them for the negative VOT case. $\boldsymbol{\phi}^+$ consists of the feature maps described in [7], and $\boldsymbol{\phi}^-$ consists of the maps described below.

We denote the time interval $[t_1, t_2]$ by $T_{t_1}^{t_2}$. Also, we define $\Delta_t^s(x^d)$ to be an approximation of the derivative of acoustic feature d at frame t, as the difference between the mean of x^d over T_t^{t+s} and the mean over T_{t-s}^t. The following feature maps make up $\boldsymbol{\phi}^-$:

- $\log E_{\text{total}}$, $\log E_{\text{low}}$, $\log E_{\text{high}}$, H_{wiener}, V evaluated at t_v and t_b (10 functions)
- $\Delta_{t_b}^s(x^d)$ for $s \in \{5, 10, 15\}$, and $d \in \{\log E_{\text{total}}, \log E_{\text{low}}, \log E_{\text{high}}, \log H_{\text{wiener}}\}$ (12 functions)
- $\Delta_{t_v}^s(x^d)$ for $s \in \{5, 10, 15\}$ and $d \in \{\log E_{\text{total}}, \log E_{\text{high}}, \log H_{\text{wiener}}\}$ (9 functions)
- Mean of $\Delta_t^s(x^d)$ over $T_{t_v}^{t_b}$ for $d \in \{\log E_{\text{low}}, \log E_{\text{high}}, \log H_{\text{wiener}}\}$ and $s \in \{5, 10\}$ (6 functions)
- Mean of V over $T_{t_v-15}^{t_v}$ and $T_{t_v}^{t_b}$ (2 functions)
- Mean of $\log E_{\text{total}}$ over $T_{t_v}^{t_b-10}$, $T_{t_b}^{t_b+50}$, and $T_{t_v}^{t_v+15}$ (3 functions)
- Difference of the mean of V, E_{high}, and H_{wiener} over $T_{t_v}^{t_b-10}$ and over $T_{t_v-50}^{t_v-5}$ (3 functions)
- Difference of the mean of V, E_{low}, E_{high}, and H_{wiener} over $T_{t_b}^{t_b+50}$ and over $T_{t_v}^{t_b-10}$ (4 functions)
- Difference of the maximum of E_{high} over $T_{t_v}^{t_b-5}$ and over $T_{t_v-50}^{t_v-5}$ (1 function)
- Difference of the maximum of E_{high} and H_{wiener} over $T_{t_b}^{t_b+50}$ and over $T_{t_v}^{t_b-5}$ (2 functions)
- Maximum of E_{low} over $T_{t_v}^{t_b-5}$ (1 function)

These feature maps were chosen by empirical examination of the spectra and waveform of voiced stops with negative VOTs. Compared to the feature maps in $\boldsymbol{\phi}^+$, those in $\boldsymbol{\phi}^-$ make greater use of the spectral energy features, and do not use the autocorrelation function or the zero crossing rate, which we found were not reliable cues for the location of prevoicing.

4. Experiments

The data come from a study of isolated word productions by L1 English speakers and L1 Portuguese/L2 English bilinguals [14]. We used a subset of this data consisting of 1331 word-initial voiced stops, of which 465/866 had negative/posiitve VOTs, produced by 10 speakers (3 monolingual, 7 bilingual). We performed experiments by cross validation: the data was divided into 4 folds, each of which was used as the test set for a classifier trained on the other 3 folds for 3 epochs. The cost parameters in Eqn. (1) were set to γ_0=4 and γ_m=100. We report results in two ways: the percentage of test examples where automatic and manual VOT measurements differ by less than a series of time thresholds, and the mean absolute difference of automatic and manual measurements.

Negative only: We first evaluate the algorithm's performance on prevoiced stops alone, by training and testing only on examples with negative VOTs. During training only $\mathbf{w}^-$ is updated, and $\mathbf{w}^+$ remains $\mathbf{0}$; at test time all examples are classified as negative. Row 2 of Table 1 summarizes the distribution of errors in this case; the mean absolute error is 4.4 ms. To our knowledge there is no previous work on automatic measurement of negative VOTs, nor any phonetic studies which report inter-transcriber agreement for prevoiced stops alone, so we cannot compare our results to previous work. However, 4.4 ms is within the range of values typically reported for

Table 1: Percent of automatic and manual measurements for test examples differing by less than t ms.

Train	Test	t=2	t=5	t=10	t=15	t=25	t=50
neg	neg	55.2	78.2	92.0	94.7	98.3	99.5
joint	neg (correct)	53.9	77.1	92.7	96.0	98.8	100
joint	neg (all)	49.4	70.8	85.2	88.2	91.0	95.3
pos	pos	50.4	80.0	93.4	94.2	95.4	96.7
joint	pos (correct)	53.2	84.4	97.2	98.3	98.7	99.0
joint	pos (all)	47.9	75.9	87.5	88.6	89.4	95.1

intertranscriber agreement on VOT in phonetic studies.

Joint classification: We next tested the algorithm on the full dataset of both positive and negative VOT examples. During training both $\mathbf{w}^-$ and $\mathbf{w}^+$ are updated, and at test time the algorithm decides on a sign and VOT measurement for each example.

We first discuss performance on classifying VOT sign for the test examples. 9.9% of the positive-VOT examples were misclassified as negative, and 7.5% of negative-VOT examples were misclassified as positive. We are not aware of previous work where a prevoicing detector is evaluated. We can indirectly evaluate our results relative to human performance by considering the rates at which speakers of Dutch—a language where voiced (voiceless) stops are usually realized with (without) prevoicing—misclassify word-initial voiceless stops as voiced (a), and vice versa (b). Two studies give rates of 4.3–19.0% for (a) and 13.6–20.0% for (b). [15, 16]. Relative to this rough gold standard, our algorithm performs well.

Next, we consider the distribution of automatic/manual measurement differences. Rows 4 and 7 of Table 1 summarize the error for all positive and negative test examples, including those where the sign was misclassified. Comparing to Rows 2 and 5 shows how much performance suffers due to the sign of the VOT test data being unknown. (Row 5 shows results for training and testing on positive-VOT data only, analogously to the procedure in "Negative only" above.) Error increases at all t for both positive and negative data (more for negative), but not drastically: about 2-8%. The mean absolute error increases from 4.4 to 10.1 ms for negative VOT examples, and from 7.7 to 9.7 ms for positive VOT examples. Rows 3 and 6 summarize the error just for examples where the VOT's sign was correctly predicted. Comparing to Rows 4 and 7 shows how much performance suffers due to sometimes predicting the wrong sign, about 4-10% at different t. The mean absolute error increases from 4.6 to 9.7 ms for positve-VOT data and from 4.1 to 10.1 ms for negative-VOT data. While the mean error values of 10 ms are higher than the figures usually reported for intertranscriber agreement on VOT in phonetic studies (about 2-6 ms), we note that the high mean absolute errors are largely due to huge errors on the small fraction of examples (<10%) where the wrong sign is chosen.

5. Discussion

We have presented an extension of the automatic VOT measurement method of [7] to handle both positive and negative VOT. To our knowledge, this is the first algorithm for measuring negative VOT, the first evaluation of a prevoicing detector, and the first measurement algorithm for the general case where VOT can be either positive or negative.

Acknowledgments. We thank Nattalia Paterson and Matt Goldrick for access to the data used in the experiments.

6. References

[1] P. Auzou, C. Ozsancak, R. Morris, M. Jan, F. Eustache, and D. Hannequin, "Voice onset time in aphasia, apraxia of speech and dysarthria: a review," *Clin. Linguist. Phonet.*, vol. 14, pp. 131–150, 2000.

[2] T. Cho and P. Ladefoged, "Variation and universals in VOT: evidence from 18 languages," *J. Phon.*, vol. 27, pp. 207–229, 1999.

[3] L. Lisker and A. Abramson, "A cross-language study of voicing in initial stops: acoustical measurements," *Word*, vol. 20, pp. 384–422, 1964.

[4] P. Niyogi and P. Ramesh, "The voicing feature for stop consonants: Recognition experiments with continuously spoken alphabets," *Speech Commun.*, vol. 41, pp. 349–367, 2003.

[5] V. Stouten and H. van Hamme, "Automatic voice onset time estimation from reassignment spectra," *Speech Commun.*, vol. 51, pp. 1194–1205, 2009.

[6] J. Hansen, S. Gray, and W. Kim, "Automatic voice onset time detection for unvoiced stops (/p/,/t/,/k/) with application to accent classification," *Speech Commun.*, vol. 52, pp. 777–789, 2010.

[7] M. Sonderegger and J. Keshet, "Automatic discriminative measurement of voice onset time," in *INTERSPEECH-2010*, pp. 2961–2964.

[8] C. Lin and H. Wang, "Automatic estimation of voice onset time for word-initial stops by applying random forest to onset detection," *J. Acoust. Soc. America*, vol. 130, pp. 514–525, 2011.

[9] P. M. van Alphen and R. Smits, "Acoustical and perceptual analysis of the voicing distinction in Dutch initial plosives: The role of prevoicing," *J. Phonetics*, vol. 32, pp. 455–491, 2004.

[10] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," in *Proc. NIPS 16*, 2003.

[11] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. 21st ICML*, 2004.

[12] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, 2006.

[13] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech coding and synthesis*, W. Kleijn and K. Paliwal, Eds. New York: Elsevier, 1995, pp. 495–518.

[14] N. Paterson, "Interactions in bilingual speech processing," Ph.D. dissertation, Northwestern University, 2011.

[15] L. Pols, "Three-mode principal component analysis of confusion matrices, based on the identification of Dutch consonants, under various conditions of noise and reverberation," *Speech Commun.*, vol. 2, pp. 275–293, 1983.

[16] R. Smits, N. Warner, J. McQueen, and A. Cutler, "Unfolding of phonetic information over time: A database of Dutch diphone perception," *J. Acoust. Soc. Am.*, vol. 113, pp. 563–574, 2003.