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

Automatic Measurement of Voice Onset Time and Prevoicing using Recurrent Neural Networks

Yossi Adi[1], *Joseph Keshet*[1], *Olga Dmitrieva*[2], *and Matt Goldrick*[3]

[1]Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel
[2]School of Languages and Cultures, Purdue University, West Lafayette, IN, USA
[3]Department of Linguistics, Northwestern University, Evanston, IL, USA
adiyoss@cs.biu.ac.il

Abstract

Voice onset time (VOT) is defined as the time difference between the onset of the burst and the onset of voicing. When voicing begins preceding the burst, the stop is called prevoiced, and the VOT is negative. When voicing begins following the burst the VOT is positive. While most of the work on automatic measurement of VOT has focused on positive VOT mostly evident in American English, in many languages the VOT can be negative. We propose an algorithm that estimates if the stop is prevoiced, and measures either positive or negative VOT, respectively. More specifically, the input to the algorithm is a speech segment of an arbitrary length containing a single stop consonant, and the output is the time of the burst onset, the duration of the burst, and the time of the prevoicing onset with a confidence. Manually labeled data is used to train a recurrent neural network that can model the dynamic temporal behavior of the input signal, and outputs the events' onset and duration. Results suggest that the proposed algorithm is superior to the current state-of-the-art both in terms of the VOT measurement and in terms of prevoicing detection.

Index Terms: voice onset time, prevoicing, recurrent neural networks

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 automatic speech recognition (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, 9, 10],[1] all making the assumption that VOT is always positive (burst onset precedes voicing onset). However, this assumption is well known to be false. 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 [11]. In other languages (e.g., Dutch, French, Spanish), voiced stops usually have negative VOT, while voiceless stops have positive VOT [12, 11].

We are aware of only a single work [13] that handles both positive and negative VOTs by extending [8]. In that work two parallel classifiers were jointly trained: one for measuring positive VOTs and one for measuring negative VOTs. The classifiers operated 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.

Current algorithms that focus on positive VOT solve two challenges in VOT measurement: detection of the onset of the burst and the onset of the voicing of the vowel. We extend these algorithms by addressing two additional challenges: determining whether or not prevoicing is present, and, when it is present, the onset of prevoicing. To simultaneously address all four challenges, we develop an algorithm that identifies up to four regions in each input utterance:

1. **Silence**: From utterance onset to prevoicing onset
2. **Prevoicing**: From prevoicing onset to burst onset
3. **Burst/Aspiration**: From burst onset to onset of voicing of vowel
4. **Vowel**: From onset of vowel voicing to end of utterance

We train a multiclass recurrent neural network to classify each frame of the input utterance as part of each region. We then use a dynamic programming algorithm to find the best segmentation of the utterance based on the classifier predictions, yielding the desired time points for calculating VOT.

Below, we outline our approach. We then assess its performance, first in the well-studied problem of positive VOT measurement and then in the less well studied case of measurement of prevoicing. We show that our algorithm out performs state-of-the-art alternatives in both cases, suggesting that it can provide a solution to the general problem of VOT measurement.

2. Problem definition

The input to our algorithm is a speech utterance containing a single stop consonant, and the output is the voice onset time (VOT), that is, the time difference between the onset of the burst and the onset of voicing. When voicing begins preceding the burst, the output is the time difference between the onset of the prevoicing and the onset of the burst. The input utterance can be of an arbitrary length, and its beginning need not be synchronized with the prevoicing (if exists), the burst onset, the

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

http://dx.doi.org/10.21437/Interspeech.2016-893

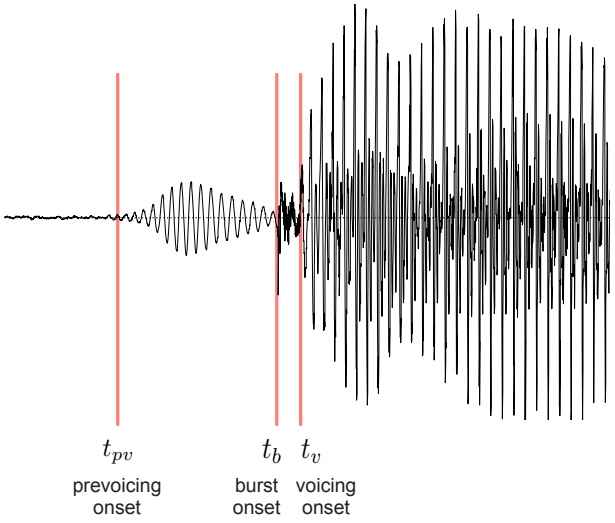

Figure 1: *Annotation example for prevoicing, burst and voicing onsets. The spoken word in this wav form is "dug."*

voicing onset, or the closure. It is required that the input utterance includes the burst, part of the vowel and the whole region of prevoicing (if exists).

Let $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ denotes the input speech utterance, 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 utterance, T, is not a fixed value since the input utterances can have different durations.

Each input utterance is associated with three elements: the prevoicing onset, $t_{pv} \in \mathcal{T}$, the onset of the burst, $t_b \in \mathcal{T}$, and the onset of the voicing of the vowel, $t_v \in \mathcal{T}$, where $\mathcal{T} = \{1, \ldots, T\}$, and $t_{pv} < t_b < t_v$. In the case of positive lag stops the prevoicing onset does not exist and t_{pv} is assigned to be -1, and the VOT is $t_v - t_b$, whereas in the case of negative lag (prevoiced) stops, all the three elements are defined and the VOT is $t_b - t_{pv}$. Our notation is depicted in Figure 1.

3. Learning apparatus

3.1. Features

Seven (D=7) acoustic features are extracted from the speech signal every 1 ms [8]. The first five features refer to an STFT taken with a 5 ms Hamming window: the total spectral energy (E_{total}), energy between 50–1000Hz (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 [14], smoothed with a 5 ms Hamming window (V).

In addition, we also use the cumulative mean, differences and max of these features similar to the feature functions used in [8] as another input to the classifier. These feature maps were chosen by empirical examination of the spectra and waveform of voiced stops with and without prevoicing. Overall we have 63 features per frame.

3.2. Recurrent neural network

One approach to determining the duration of a phonetic property is to predict at each time frame whether the property is present or absent; the predicted duration is then the smoothed, continuous set of frames where the property is likely to be present [15]. In this work we extend this method, generating predictions using a Recurrent Neural Network (RNN). This allows the prediction of whether a property is present to be sensitive to the relationship between frames.

We implement a network of two-layers of stacked LSTMs [16], which has shown considerable success in analyzing dynamic temporal behavior [17, 18]. We use an in-house implementation that is based on the Torch7 toolkit [19, 20]. Formally, the implementation is the following set of recursive equations, where the weights and the biases are denoted by $\mathbf{W}$ and $\mathbf{b}$, respectively, and σ is the sigmoid function:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}x_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \quad (4)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (5)$$

The input to the RNN classifier is a sequence of T tuples, where each tuple is composed of the acoustic features $\mathbf{x}_t$ and a corresponding label y_t from the set $\mathcal{Y}$ ={silence, prevoicing, burst, vowel} for $1 \leq t \leq T$ as follows:

$$y_t = \begin{cases} \text{silence} & 1 \leq t < t_{pv} \\ \text{prevoicing} & t_{pv} \leq t < t_b \\ \text{burst} & t_b \leq t < t_v \\ \text{vowel} & t_v \leq t \leq T \end{cases}. \quad (6)$$

We trained a multiclass RNN to predict the label of each frame, and optimized the negative log-likelihood using Adagrad [21] with learning rate of 0.1 and batch size of 32 examples. We used two dropout layers after each LSTM with dropout rate of 0.8. We stoped training the network after 5 epochs with no loss improvement on the validation set.

3.3. Inference

The Multiclass RNN outputs a probability for each class. At inference time, we use these probabilities to predict the most likely segmentation of the utterance. Since the predictions can be noisy, and we require a smooth prediction, we use a dynamic programming algorithm to infer the best segmentation. This procedure is described in Figure 2. Denote by $\hat{P}(y_t|\mathbf{x}_t)$ the predicted probability of the network for the input $\mathbf{x}_t$ and class y_t, and denote by T_m the maximum allowable size of each segment. Given $y = \mathcal{Y}$ be the events in the utterance, and two time indices $t, t' \in \mathcal{T}$, denote by $D(y, t, t')$ the score for the prefix of the events sequence: silence,$\ldots, y$, assuming that their actual onsets are $1, t_{pv}, \ldots, t'$, and assuming that $y_{i+1} = t$. The best sequence of actual onsets is obtained from the algorithm by saving the intermediate values that maximize each expression in the recursion step.

4. Experiments

In order to have better understanding on the capabilities of the proposed model we divide the analyses into two sections. First,

Initialization:

for y =[silence]

$$D_{\text{neg}}(y,t,0) = \hat{P}(y|\mathbf{x}_t) \quad 1 \le t \le T_m$$

$$D_{\text{pos}}(y,t,0) = \hat{P}(y|\mathbf{x}_t) \quad 1 \le t \le T_m$$

Recursion:

for y =[prevoicing, burst, vowel]

$$D_{\text{neg}}(y,t',t'') = \max_{t'''} \sum_{t=t''}^{t'} \hat{P}(y|\mathbf{x}_t) + D_{\text{neg}}(y{-}1,t''',t'')$$

for y =[burst, vowel]

$$D_{\text{pos}}(y,t',t'') = \max_{t'''} \sum_{t=t''}^{t'} \hat{P}(y|\mathbf{x}_t) + D_{\text{pos}}(y{-}1,t''',t'')$$

Termination: for y =[vowel]

$$D^\star = \max_{t'} \left\{ D_{\text{neg}}(y,T,t'), D_{\text{pos}}(y,T,t') \right\}$$

Figure 2: *Dynamic programming algorithm for post-process inference.*

we trained the network to measure only positive VOT and compared it to the current state of the art algorithm. We then trained the network to measure positive and negative VOT jointly and compared it to the current state of the art algorithm.

4.1. Positive VOT

To evaluate the performance of our model in measuring positive VOT we used data from 9 speakers drawn from the Northwestern University community [22]. Participants read aloud tongue twisters consisting of alternating pairs of voiced and voiceless consonants (e.g., *pin bin bin pin*). Recordings were randomly assigned to two highly trained coders. VOT was coded via inspection of the waveform, from burst to onset of periodicity in the vowel. Reliability (n = 257 tokens from 5 participants) was very high (*r = 0.996*).

We trained the network on data from 4 speakers (7,654 acoustic segments), with 15% from the data for validation, and tested on data from the remaining 5 speakers (8,628 acoustic segments). Overall we used 504,790 frames for training, 89,080 frames for validation and 143,458 frames for test. The dataset is roughly balanced with respect to the number of VOT and none-VOT frames. We denote our system as *DeepVOT*. The same dataset with the same data split was used to train the algorithm in [8], denoted *AutoVOT*. Table 1 summarizes the distribution of automatic/manual differences over the test set.

Results suggests that our algorithm is superior to the AutoVOT algorithm; DeepVOT exhibits smaller deviations from manual measurements. This is a non-trivial improvement, especially when the tolerance value, t, is small, i.e. 2 or 5 msec.

To see if the system suffers a decline in results when using a model that was trained on one dataset but tested on a different one, we evaluated this trained DeepVOT system on a new data set. We examined positive-lag VOTs from 16 native English speakers at Purdue University who read aloud a list of printed words three times. Recordings were randomly assigned to four trained coders. The VOT intervals were coded via inspection of the waveform and the spectrogram of word- initial stops. Positive VOT was measured from the onset of burst until the onset

Table 1: *Proportion of differences between automatic and manual measures falling at or below a given tolerance value (in msec). For example, for* DeepVOT*, in 75.3% of examples in the test set the difference between automatic and manual measurements was 2 msec or less.*

Model		$t \le 2$	$t \le 5$	$t \le 10$	$t \le 15$	$t \le 25$	$t \le 50$
AutoVOT	mean	50.5	79.1	91.7	94.4	96.8	98.8
	std	4.5	4.7	2.6	1.9	1.2	0.6
DeepVOT	mean	75.3	91.9	95.9	97.1	98.2	99.1
	std	9.4	3.4	1.6	1.1	0.9	0.7

Table 2: *Performance when the system was trained on data from participants at Northwestern University and tested on a second dataset from Purdue University. Proportion of differences between automatic and manual measures falling at or below a given tolerance value (in msec).*

Type	$t \le 2$	$t \le 5$	$t \le 10$	$t \le 15$	$t \le 25$	$t \le 50$
Voiced	63.1	91.9	96.9	98.3	99.3	100
Voiceless	56.5	81.6	86.8	87.2	87.3	89.0

of periodicity in the vowel. All segmentations were inspected by a fifth, highly trained coder and corrected if needed. It can be seen from Table 2 that system performance was quite high even when testing on a novel dataset.

4.2. Negative VOT (prevoicing)

Next, we investigated the performance of our algorithm regarding negative VOT (prevoicing) measurement. We use the data set from a study of isolated word productions in picture naming and reading aloud by L1 English speakers and L1 Portuguese/L2 English bilinguals from the Northwestern University community [23]. All tokens were measured by one highly trained coder. Prevoicing, burst, and onset of periodicity in the vowel were coded via inspection of the waveform. Reliability was assessed by a second trained coder who measured 958 tokens; agreement was very high (*r = 0.972*).

We used a subset of this data consisting of 1446 word-initial voiced stops produced by 10 speakers (3 monolingual, 7 bilingual), evenly split between prevoiced and short-lag VOT. We used 1074 acoustic segments for a training set, with 15% of these used as validation set (146,254 frames for training set, 25,809 frames for validation set). The test set contained 372 acoustic segments (60,881 frames). Prevoiced and short-lag were evenly sampled in training, test and validation sets.

The network did extremely well at detecting prevoicing, with accuracy rate of 97.8%, precision rate of 95.9% and recall rate of 100%. To evaluate performance in measuring VOT, we report results of the percentage of test examples where automatic and manual VOT measurements differed by less than a series of time thresholds. For this analysis, in cases where the manual and network disagree in the presence of prevoicing, the duration of VOT was set by the following rule:

- If the network classifies the input as negative VOT, but the manual annotation was positive, we consider the pre-

Table 3: *Performance on dataset including prevoicing. Proportion of differences between automatic and manual measures falling at or below a given tolerance value (in msec), where* (c) *and* (a) *stand for* correct *and* all*, respectively.*

Model	Type	$t \leq 2$	$t \leq 5$	$t \leq 10$	$t \leq 15$	$t \leq 25$	$t \leq 50$
AutoVOT	neg (c)	53.9	77.1	92.7	96.0	98.8	100
	neg (a)	49.4	70.8	85.2	88.2	91.0	95.3
	pos (c)	53.2	84.4	97.2	98.3	98.7	99.0
	pos (a)	47.9	75.9	87.5	88.6	89.4	95.1
DeepVOT	neg (c)	63.5	78.1	91.0	95.0	98.9	100
	neg (a)	60.7	75.8	89.8	94.6	98.4	100
	pos (c)	80.1	95.7	98.4	98.9	100	100
	pos (a)	80.1	95.7	98.4	98.9	100	100

voicing duration as the VOT.

- If the network classifies the input as positive VOT, but the manual annotation was negative, we consider the burst duration as the VOT.

We compared our result to the state-of-the-art results on this dataset, reported in [13], provide a baseline for the DeepVOT algorithm's performance. The results are summarized in Table 3.

5. Discussion

We have presented a new system for detecting positive and negative VOTs. Our method is based on sequential deep learning, which allows us to use the same learning framework and the same set of feature set for measuring both positive and negative VOTs. For future work we would like to explore the option of optimizing the network end-to-end including the dynamic programming post-processing. Such optimization may further improve the accuracy of such networks.

This approach opens up the possibility of extending automatic analysis of VOT beyond prototypical English productions to cover the many languages that consistently utilize prevoicing. DeepVOT will be publicly available at `https://github.com/MLSpeech/DeepVOT`.

6. Acknowledgements

Research supported in part by NIH grant 1R21HD077140.

7. 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 *Proc. of Interspeech*, 2010, pp. 2961–2964.

[8] ——, "Automatic measurement of voice onset time using discriminative structured predictiona)," *The Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 3965–3979, 2012.

[9] 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.

[10] A. Prathosh, A. Ramakrishnan, and T. Ananthapadmanabha, "Estimation of voice-onset time in continuous speech using temporal measures," *The Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. EL122–EL128, 2014.

[11] O. Dmitrieva, F. Llanos, A. A. Shultz, and A. L. Francis, "Phonological status, not voice onset time, determines the acoustic realization of onset f0 as a secondary voicing cue in spanish and english," *Journal of Phonetics*, vol. 49, pp. 77–95, 2015.

[12] 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.

[13] K. Henry, M. Sonderegger, and J. Keshet, "Automatic measurement of positive and negative voice onset time." in *INTERSPEECH*, 2012, pp. 871–874.

[14] 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.

[15] Y. Adi, J. Keshet, and M. Goldrick, "Vowel duration measurement using deep neural networks," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[18] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[19] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[20] N. Léonard, S. Waghmare, and Y. Wang, "rnn: Recurrent library for torch," *arXiv preprint arXiv:1511.07889*, 2015.

[21] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[22] M. Goldrick, H. R. Baker, A. Murphy, and M. Baese-Berk, "Interaction and representational integration: Evidence from speech errors," *Cognition*, vol. 121, no. 1, pp. 58–72, 2011.

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