# Using Syllable Nuclei Locations to Improve Automatic Speech Recognition in the Presence of Burst Noise

*Chris D. Bartels and Jeff A. Bilmes*

University of Washington
Department of Electrical Engineering
Seattle, WA 98195
{bartels,bilmes}@ee.washington.edu

10.21437/Interspeech.2008-130

## Abstract

In this work we combine a conventional phone-based automatic speech recognizer with a classifier that detects syllable locations. This is done using a dynamic Bayesian network. Using oracle syllable detections we achieve a 17% relative reduction in word error rate on the 500 word task of the SVitchboard corpus. Using estimated locations we achieve a 2.1% relative reduction which is significant at the 0.02 level. The improvement in the estimated case is from reducing insertions caused by burst noise.

**Index Terms**: Automatic speech recognition, dynamic Bayesian networks, syllables, speaking rate

## 1. Introduction

In many cases automatic speech recognizers will incorrectly decode burst noises as speech. These noises, such as cross-talk, are neither speech nor silence, and many particular noises may never occur in the training data. Non-speech noise is not explicitly labeled and whenever a portion of the speech is labeled “silence” it may also include noise. Here we propose a method that improves on burst noise identification using a dynamic Bayesian network that combines a conventional phone-based automatic speech recognizer with a classifier that identifies locations of syllable nuclei.

Perceptual experiments have given evidence that speech intelligibility depends on syllable-length modulations in the speech signal [2, 3, 4]. This has led a number of researches to incorporate syllable information into automatic speech recognition (ASR). In [5] it was proposed that syllables should be used as a basic recognition unit. This work proposed that, instead of phones, speech should be characterized in terms of syllable onset, nucleus, and coda along with sub-classes within each of these sub-syllable units. This was implemented at a later date in [6]. In [7], a phone-based recognizer is combined with an asynchronous syllable-based recognizer using what were called “recombination states”, and in [8, 9] phone and syllable based recognizers are combined using N-Best lists. This work has much in common with the ideas presented in [10] and [11, 9]. In [10], speech was segmented using syllable onset estimations for use in template matching. In [11, 9] syllable onsets are detected using a neural network, and hypotheses that are inconsistent with these detections are removed from a lattice.

There has been much research in the area of noise robust ASR. A somewhat dated but excellent survey can be found in [12]. One set of approaches, such as [13], consists of preprocessing the features in a way that makes them more robust to noise. These approaches tend to be more effective on stationary noise than on bursts. Some voice activity detectors (VADs) can handle bursts by building models for speech, noise, and silence. If one has instances of many burst noises at training time a noise model can be trained using a hidden Markov model (HMM) as in [14]. The output of a neural network based VAD was integrated into ASR in [15]. This VAD made binary speech/non-speech decisions but was integrated into recognition in a soft manner using a penalty factor. A noise model is built in an unsupervised manner using a switching Kalman filter in [16]. Unlike the HMM and neural network, this method can handle noises not seen in training. The Kalman filter uses standard ASR analysis windows and has only been integrated into ASR by making “hard” segmentation decisions. The model proposed here makes use of temporal dynamics at a syllable-length time scale, and it makes the speech/non-speech decision in a soft manner at recognition time. It makes use of a trained silence model and at the same time is theoretically robust to noises not seen in training.

This work builds on the ideas reported by these authors in [1]. In the previous work, when using estimated syllable nuclei a small improvement was shown on the 10 word task from [17] and no improvement was shown on the 500 word task. Here we report an improved baseline result and an improvement on the 500 word task. Previously the word and syllable detection streams were synchronized at the ends of words and utterances, and the newly proposed graph synchronizes them at the end of every syllable. The nuclei features in the previous work were from the correlation-based method given in [18, 19]. This paper uses neural network derived features that will be described in Section 2. Finally, in this work the possible nuclei are selected off-line and a binary feature is given to the model. In the previous work, each potential nucleus had an associated floating point number indicating a confidence that the maximum was actually a nucleus, and a “soft” selection was incorporated into the DBN. A soft selection could be added into the model presented here, but empirically we found the binary feature to perform better.

## 2. Detecting Syllables

English syllables are typically defined by an onset, nucleus, and coda. This structure generally corresponds to the rising and falling of sonorance and energy in the speech signal. The syllable nucleus is always a vowel and is the most sonorant portion of the syllable. Syllables may or may not have an onset and/or a coda. If the onset and coda do exist, they consist of consonants with the most sonorant consonants directly surrounding the nucleus. The consonants become decreasingly sonorant as one looks towards the beginning and end of the syllable.

There have been two primary ways that syllables have been identified in other work. The first is to train a discriminative classifier to locate syllable boundaries or nuclei. In [11, 9] a neural network was used to find syllable onsets. Temporal Flow Model neural networks were used in [20] to find syllable boundaries. In [21] a support vector machine was used to classify speech as sonorant or non-sonorant.

In other cases, signal processing methods have been used to identify syllables. This class of methods was developed for the purpose of estimating speaking rate. A spectral correlation metric was used to identify nuclei in [22], and this method was improved upon in [18, 19]. In [23] syllable nuclei are detected using a modified loudness function combined with the zero-crossing rate.

Our method uses the posterior of a neural network as a measure of sonorance and interprets the peaks in this posterior as syllable nuclei. We make use of an existing set of publicly available neural networks trained for phone classification. A detailed description of the networks can be found in [24].

The inputs to the neural network are nine frames of PLP cepstra plus energy along with their deltas and double deltas. The features are calculated every 10ms with 25ms windows, and are mean and variance normalized on a per speaker basis. These neural networks produce posteriors for 46 phones. We divide the phones into three classes: vowels, consonants, and silence. The posteriors for individual vowels are summed to give a single overall vowel posterior. The silence posterior is used directly, and the posteriors for the consonants are not used.

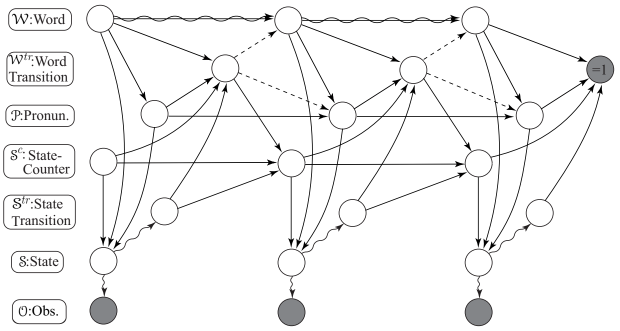
For each utterance, the vowel and silence posteriors are smoothed in time using a 9 frame Hamming window. The length of the window was chosen by the recognition performance of the resulting features on the development set. Next, the maxima in the smoothed vowel posterior are found. A maximum is taken to be any frame with a posterior larger than its two adjacent frames. Maxima that occur less than 5 frames after a previous maximum are thrown away. If a maximum is found at a point where the smoothed silence posterior is greater than 0.5, that maximum is also thrown away. These remaining maxima are taken to be estimated locations of the syllable nuclei. We then create a binary feature for each time frame. These features equal 1 in the frames where a maximum occurred, and they equal 0 in all other frames.

We also present results using *Oracle* syllable features. To obtain these, the baseline recognizer is used along with the time-aligned transcriptions to create time-aligned phone transcriptions of each utterance. The beginnings and ends of all words are marked as syllable boundaries. It is assumed that there is one syllable per vowel, and in multiple syllable words within-word syllable boundaries are marked using a heuristic. This heuristic splits strings of consonants that occur between vowels by placing the first consonant in the coda of the first syllable and subsequent consonants in the onset of the second syllable. The oracle syllable nuclei are set to be the points in the centers of the syllable boundaries, and binary features are created in the same manner as the estimated features.

## 3. Model

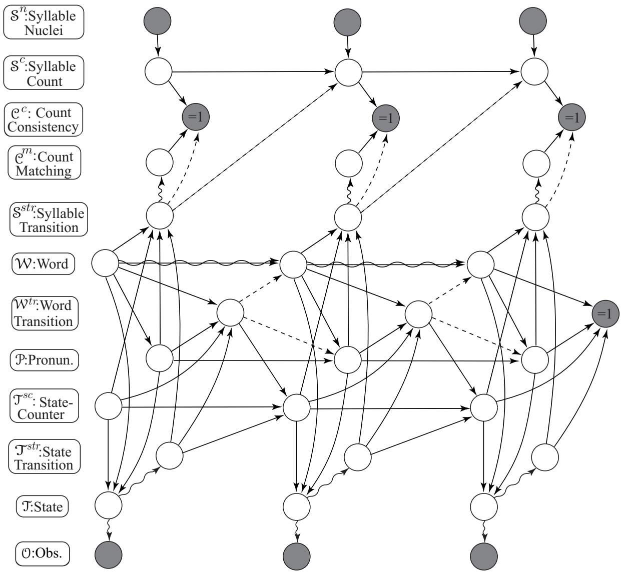
The baseline system is a conventional HMM implemented using the DBN shown in Figure 1. This DBN was developed for [25]. For more on DBNs in automatic speech recognition see [27, 26]. This graph uses state clustered within-word triphones and implements a three state left-to-right topology.

The model introduced in this paper is called **Syllable-Level** and is given in Figure 2. The lower portion of the graph is identical to the baseline model. The upper portion of the model counts the number of detected syllables since the beginning of each hypothesized syllable or silence region. We expect to count 1 detection for each syllable, and we expect to count 0 detections in a silence or short pause. Whenever the lower portion reaches the end of a syllable or silence region, an additional probability is multiplied into the hypothesis score. This additional factor is the probability of the current detection count given the number of detections we expect to see. This distribution is learned from the training data. Note that the syllable detection portion of the model adds additional state. This state allows the model to give differing scores depending on how the beginnings and ends of the hypothesized words align with the stream of detected syllables. One view of this system is that the phone-based recognizer hypothesizes a particular number of syllables, the syllable detector hypothesizes another number, and the graphical model provides probabilistic "glue" that encourages consistency between the two.



$\mathcal{W}$:Word
$\mathcal{W}^{tr}$:Word Transition
$\mathcal{P}$:Pronun.
$\mathcal{S}^{c}$: State-Counter
$\mathcal{S}^{tr}$:State Transition
$\mathcal{S}$:State
$\mathcal{O}$:Obs.
=1

Figure 1: Baseline Model [25, 26]. This is a standard speech HMM represented as a DBN. Hidden variables are white while observed variables are shaded. Straight arrows represent deterministic relationships, curvy arrows represent probabilistic relationships, and dashed arrows are switching relationships.



$\mathcal{S}^{n}$:Syllable Nuclei
$\mathcal{S}^{c}$:Syllable Count
$\mathcal{C}^{c}$: Count Consistency
$\mathcal{C}^{m}$:Count Matching
$\mathcal{S}^{str}$:Syllable Transition
$\mathcal{W}$:Word
$\mathcal{W}^{tr}$:Word Transition
$\mathcal{P}$:Pronun.
$\mathcal{T}^{sc}$: State-Counter
$\mathcal{T}^{str}$:State Transition
$\mathcal{T}$:State
$\mathcal{O}$:Obs.
=1

Figure 2: Syllable-Level graph (see Figure 1 for key). The bottom portion of this graph is identical to the baseline model given in Figure 1. The upper portion of the model keeps track of the number of estimated syllable detections that occur during the duration of each syllable hypothesized by the lower portion of the model.

The variables in the syllable detection portion of the graph will now be described in detail. The variable *Syllable Transition*, notated $\mathcal{S}^{str}$, can take on four values: $syllable$, $silence$, $sp$, and $none$. It is equal to $syllable$ when a hypothesized word is in the last frame of a syllable, $silence$ when it is in the last frame of a hypothesized silence, $sp$ when it is in the last frame of a hypothesized short pause, and $none$ when it is not in the last frame of a syllable or silence region. The variable *Syllable Nuclei* is a binary observation using the syllable detection features described in the previous section. The variable *Syllable Count* keeps a count of the number of detected syllables since the beginning of each hypothesized syllable (this is a positive count, starting from zero). The count is reset to zero whenever *Syllable Transition* is equal to $syllable$, $silence$, or $sp$. *Count Matching*, $\mathcal{C}^m$, is a random variable that gives a distribution over detected syllable counts given the *Syllable Transition*. Both *Syllable Count* and *Count Matching* have 4 possible values. These represent a total count of 0, 1, 2, and $\geq 3$. *Count Consistency* is a constraint that is enforced whenever *Syllable Transition* does not equal $none$. When the constraint is turned on, it forces *Syllable Count* to be equal to *Count Matching*. This will cause $p(\mathcal{C}^m = c|\mathcal{S}^{str} = s)$ to be multiplied into the hypothesis score. Whenever *Syllable Transition* equals $none$ and the constraint is turned off, $p(\mathcal{C}^m = c|\mathcal{S}^{str} = s)$ has no effect on the model.

The Gaussian parameters and transition probabilities were trained for the baseline model with expectation maximization (EM). These parameters were imported directly into the Syllable-Level model and held fixed while training the Syllable-Level model's parameters . The only distribution in Syllable-Level that needs to be trained is $p(\mathcal{C}^m|\mathcal{S}^{str})$. This training converges with four additional EM iterations. The language model scale and word insertion penalty is determined by evaluating the recognition performance over a range of settings on the development set. The Syllable-Level model has an additional scaling factor on $p(\mathcal{C}^m|\mathcal{S}^{str})$. This scale along with the language model scale and word insertion penalty are optimized on the development set separately from the baseline.

## 4. Experiments and Results

All experiments were performed on the 500 word task of the SVitchboard corpus [17]. SVitchboard is a small, closed vocabulary subset of Switchboard I [28]. This allows experimentation on spontaneous continuous speech, but with less computational complexity and experiment turn-around time than a true large vocabulary task. The A, B, and C folds were used for training, the D_short fold was used as the development set, and the E fold was used as the evaluation set. The observation vectors are 13 dimensional PLPs normalized on a per conversation side basis along with their deltas and double-deltas. All models were trained and decoded using The Graphical Models Toolkit (GMTK) [29].

The neural networks (from [24]) that are used to calculate the vowel posteriors for the syllable detector were trained on 2000 hours of Fisher data. Although this data is not as well matched as the in-domain data, it is important to note that the syllable detection likely benefits from being trained on a large amount of data outside the SVitchboard corpus.

Results for all experiments are given in Table 1. Using the oracle features we achieve a 17% relative reduction in word error rate. When using the estimated syllable locations we achieve a 2.1% relative reduction which is significant at the 0.02 level according to a difference of proportions significance test.

Table 1 also presents a result, labeled **Silence Oracle**, where the baseline and Syllable-Level models are told which frames should be decoded as silences. Note that the "silence" regions also include the non-speech noise, and with the help of the oracle information neither system will ever hypothesize speech in a noise region. In this case, the Syllable-Level model does not show any improvement over the baseline. This result has two implications. First, it shows that the improvement in the Syllable-Level model primarily comes from the reduction of insertions due to noise. Second, the absolute improvement of 1.1% using the estimated features is 42% of the possible improvement of 2.6% given by the silence oracle.

The Syllable-Level model is more computationally expensive than the baseline model. Currently it runs two to three times slower than the baseline, but we expect to be able to speed up its performance by optimizing the beam settings and the graph triangulation.

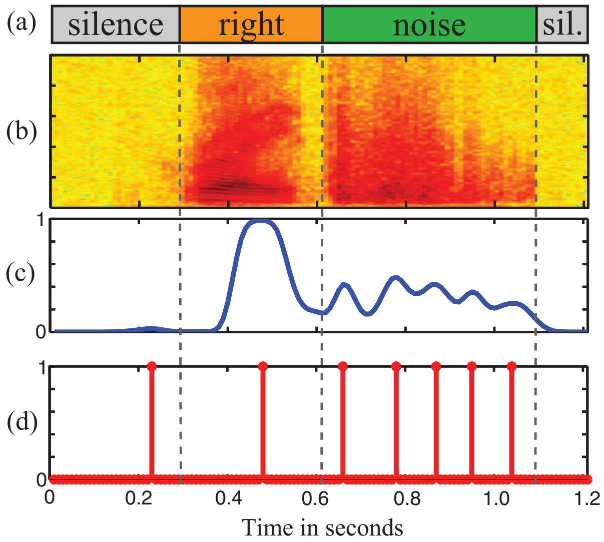


Figure 3: Example to illustrate the syllable detection features. (a) Gives the transcription. Note that the baseline recognizer decodes the noise as an additional "right", but Syllable-Level correctly decodes the noise as "silence". (b) Spectrogram of the audio signal. (c) Smoothed posterior from the neural network vowel detector (d) Maxima in the vowel posterior are interpreted as syllable nuclei. The one syllable in "right" is detected correctly. There are false detections in the silence and noise portions.

## 5. Discussion

When a noise occurs, often both the speech recognition unit and the neural network give a high posterior probability to speech. The Syllable-Level model is able correct mistakes in the baseline because it recognizes that a mismatch between the baseline hypothesis and the detected number of syllables is indicative of noise. Figure 3 gives an example of the Syllable-Level graph removing such an error. The baseline model correctly decodes the word "right", but then incorrectly inserts a second "right" during a breath noise. The breath noise does not sound like the word "right", but it has an even worse acoustic match to the silence model. The syllable detector correctly identifies the syllable nucleus for the word "right", but it has five false detections during the breath noise. Syllable-Level will give a low probability to decoding "right" during the noise because it is not a five syllable word. There is also a false detection during the initial silence, but it does not affect the result. Many of the false detections are clearly not syllable nuclei and one could easily suppress many of these based on the vowel or silence posteriors at these points. The primary reason that this is not done is that the frequency of their occurrence is informative, and including them improves the speech detection performance. In addition, removing detections based on the posterior can be problematic because the neural network is often fooled by the same noises that fool the baseline recognizer. In Figure 3, the five false detection points during the breath noise have a vowel posterior that is larger than the silence or consonant posteriors. Although these posteriors are smaller than the one correct detection in the figure, in other speech segments it is not uncommon to see true vowels with similarly low posteriors.

The corrections mentioned above (and similar such corrections) are not a result of a particularly good acoustic match between training and test data. Instead, they are a result of two portions of our model being inconsistent and thereby precluding "speech" as being hypothesized at those points. Furthermore, theoretically there do not need to be any examples of a specific

| | | Development | | | | Evaluation | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **S** | **D** | **I** | **WER** | **S** | **D** | **I** | **WER** |
| **Baseline** | | 595 | 195 | 117 | 49.5% | 6799 | 2547 | 1124 | 52.3% |
| **Syllable-Level** | *Oracle Features* | 633 | 26 | 53 | 38.8% | 7376 | 474 | 799 | 43.2% |
| | *Estimated Features* | 564 | 219 | 86 | 47.4% | 6684 | 2752 | 813 | 51.2% |
| **Baseline + Silence Oracle** | | 600 | 189 | 65 | 46.6% | 6739 | 2478 | 726 | 49.7% |
| **Syllable-Level + Silence Oracle** | *Estimated Features* | 626 | 137 | 91 | 46.6% | 7048 | 1959 | 969 | 49.8% |

Table 1: Table of Results. S, D, and I are counts of substitutions, deletions, and insertions. WER is percent word error rate.

noise in the training data for a mismatch to occur and cause it to be properly decoded as non-speech. (This hypothesis has not been directly tested, though. Noises are not marked in our corpus making such a claim difficult to verify empirically.) Our method can also be seen as classifier combination. Typical classifier combination approaches concentrate on ways of choosing between a set of alternative hypotheses. The effect seen here is different in that neither hypothesis is correct, instead the mismatch is used as an information source. The switching Kalman filter based VAD in [16] can also deal with unseen noises, but it relies on building an accurate noise model in an unsupervised manner.

Another important aspect to this work is its ability to incorporate information over longer time spans than typical ASR systems. This is done in two ways. First, locating peaks in the smoothed neural network posterior analyzes the signal over syllable length time scales. Second, the syllable counting portion of the model adds additional state that is related to the duration of each hypothesized syllable. Other work that makes use of longer time scales is typically in the form of a feature that is appended to the standard observation vector, such as [30].

In conclusion, the Syllable-Level model successfully is able to discriminate between speech and non-speech noise which results in a significant reduction in word error rate. Currently, the model does not appear to provide any advantage inside sub-segments that the baseline model correctly hypothesizes as speech. Given the large improvement using the oracle syllable nuclei, future work will examine how to make the syllable detection more robust. This framework could also be used to incorporate other prosodic detections. In particular, a logical extension would be to add information distinguishing between stressed an unstressed syllables.

**Acknowledgments:** This work was supported by ONR MURI grant N000140510388 and by NSF grant IIS-0093430.

## 6. References

[1] C. Bartels and J. Bilmes, "Use of syllable nuclei locations to improve ASR," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Kyoto, Japan, December 2007.

[2] R. Drullman, J. M. Festen, and R. Plomp, "Effect of temporal envelope smearing on speech reception," *Journal of the Acoustical Society of America*, vol. 95, no. 2, p. 10531064, February 1994.

[3] T. Arai, M. Pavel, H. Hermansky, and C. Avendano, "Intelligibility of speech with filtered time trajectories of spectral envelopes," in *Proc. of ICSLP*, vol. 4, 1996, pp. 2490–2493.

[4] S. Greenberg, T. Arai, and R. Silipo, "Speech intelligibility derived from exceedingly sparse spectral information," in *Proc. of ICSLP*, 1998, pp. 74–77.

[5] O. Fujimura, "Syllable as a unit of speech recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 82–87, Feburary 1975.

[6] P. Green, N. R. Kew, and D. A. Miller, "Speech representations in the SYLK recognition project," in *Visual Representation of Speech Signals*, M. Cooke, S. Beet, and M. Crawford, Eds. John Wiley & Sons, 1993, ch. 26, pp. 265–272.

[7] S. Dupont, H. Bourlard, and C. Ris, "Using multiple time scales in a multi-stream speech recognition system," in *Proc. of Eurospeech*, 1997.

[8] S.-L. Wu, E. E.D. Kingsbury, N. Morgan, and S. Greenberg, "Incorporating information from syllable-length time scales into automatic speech recognition," in *Proc. of ICASSP*, 1998.

[9] S.-L. Wu, "Incorporating information from syllable-length time scales into automatic speech recognition," Ph.D. dissertation, University of California, Berkeley, Spring 1998.

[10] M. Hunt, M. Lennig, and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," in *Proc. of ICASSP*, 1980.

[11] S.-L. Wu, M. Shire, S. Greenberg, and N. Morgan, "Integrating syllable boundary information into speech recognition," in *Proc. of ICASSP*, 1997.

[12] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, no. 3, pp. 261–291, April 1995.

[13] C.-P. Chen, J. Bilmes, and D. Ellis, "Speech feature smoothing for robust ASR," in *Proc. of ICASSP*, 2005, pp. 525–528.

[14] T. Pfau, D. Ellis, and A. Stolcke, "Multispeaker speech activity detection for the icsi meeting recorder," in *Proc. of ASRU*, 2001.

[15] F. Beaufays, D. Boies, M. Weintraub, and Q. Zhu, "Using speech/non-speech detection to bias recognition search on noisy data," in *Proc. of ICASSP*, 2003.

[16] M. Fujimoto and K. Ishizuka, "Noise robust voice activity detection based on switching kalman filter," *IEICE Trans. on Information & Systems*, vol. E91-D, no. 3, pp. 467–477, 2008.

[17] S. King, C. Bartels, and J. Bilmes, "SVitchboard: Small-vocabulary tasks from switchboard," in *Proc. of Eurospeech*, 2005.

[18] D. Wang and S. Narayanan, "Speech rate estimation via temporal correlation and selected sub-band correlation," in *Proc. of ICASSP*, 2005.

[19] ——, "Robust speech rate estimation for spontaneous speech," *IEEE Trans. on Speech, Audio and Language Processing*, 2007.

[20] L. Shastri, S. Chang, and S. Greenberg, "Syllable detection and segmentation using temporal flow neural networks," in *Proc. of the 14th International Congress of Phonetic Sciences*, 1999.

[21] K. Schutte and J. Glass, "Robust detection of sonorant landmarks," in *Proc. of Eurospeech*, 2005, pp. 1005–1008.

[22] N. Morgan and E. Fosler-Lussier, "Combining multiple estimators of speaking rate," in *Proc. of ICASSP*, 1998.

[23] T. Pfau and G. Ruske, "Estimating the speaking rate by vowel detection," in *Proc. ICASSP*, vol. 2, 1998, pp. 945–948.

[24] J. Frankel, M. Magimai-Doss, S. King, K. Livescu, and O. Çetin, "Articulatory feature classifiers trained on 2000 hours of telephone speech," in *Proc. of Interspeech*, Antwerp, Belgium, 2007.

[25] K. Livescu et al., "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU summer workshop," in *Proc. of ICASSP*, 2007.

[26] J. Bilmes and C. Bartels, "A review of graphical model architectures for speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 89–100, September 2005.

[27] G. Zweig, "Speech recognition with dynamic Bayesian networks," Ph.D. dissertation, University of California, Berkeley, Spring 1998.

[28] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. of ICASSP*, 1992.

[29] J. Bilmes, *GMTK: The Graphical Models Toolkit*, 2002.

[30] H. Hermansky and S. Sharma, "TRAPs - Classifiers of temporal patterns," in *Proc. of ICSLP*, 1998.