Speech Activity Detection on YouTube Using Deep Neural Networks

Neville Ryant, Mark Liberman, Jiahong Yuan

Linguistic Data Consortium, PA, USA

{nryant,markyliberman,jiahong.yuan}@gmail.com

Abstract

Speech activity detection (SAD) is an important first step in speech processing. Commonly used methods (e.g., frame-level classification using gaussian mixture models (GMMs)) work well under stationary noise conditions, but do not generalize well to domains such as YouTube, where videos may exhibit a diverse range of environmental conditions. One solution is to augment the conventional cepstral features with additional, hand-engineered features (e.g., spectral flux, spectral centroid, multiband spectral entropies) which are robust to changes in environment and recording condition. An alternative approach, explored here, is to learn robust features during the course of training using an appropriate architecture such as deep neural networks (DNNs). In this paper we demonstrate that a DNN with input consisting of multiple frames of mel frequency cepstral coefficients (MFCCs) yields drastically lower frame-wise error rates (19.6%) on YouTube videos compared to a conventional GMM based system (40%).

Index Terms: speech activity detection, voice activity detection, segmentation, deep neural networks

1. Introduction

Speech Activity Detection (SAD), the process of identifying all speech containing segments in an audio signal, is an important first step for a number of speech processing applications. Among other benefits, accurate SAD greatly speeds up manual transcription [1] and reduces error rates and overall computation time for speech recognition [2] and speaker recognition [3]. SAD may also aid human listeners by reducing the cognitive overhead associated with monitoring for speech containing regions in signals with low a priori probabilities of speech.

A number of techniques have been proposed for SAD, including both unsupervised systems that threshold against the value of some energy or voicing feature [4] and supervised systems which train a classifier with features such as Mel-frequency cepstral coefficients (MFCCs) or perceptual linear prediction coefficients (PLPs) as input. Amongst systems in the latter class, support vector machines [5]), gaussian mixture models (GMMs) [6], and multi-layer perceptrons [6] have all found success. Algorithms for structured prediction have also found success, including both hidden markov models (HMMs) [2] and conditional random fields (CRFs) [7].

In recordings where the ratio of speech to non-speech signals is reasonably high, or where the non-speech background is relatively stable, such methods work quite well. Frame-wise error rates of less than 10% (relative to human annotation) are commonly reported [2, 7, 6]. However, for recordings with low SNR and/or diverse kinds of non-speech background, the performance of these techniques may be much lower. The audio tracks of web video recordings, as a group, are challenging in this way and reported error rates on such tasks have been much higher than for their clean counterparts [8, 9].

One approach to SAD on this challenging domain, explored by [9], utilizes hand-crafted sets of features purported to be robust to changes in environment and recording conditions. An alternative approach, explored here, is to learn robust features during the course of training using an appropriate architecture such as deep neural networks (DNNs) [10].

2. Evaluation details

2.1. Training/test sets

We make use of 65 hours of web videos selected from the HAVIC corpus [11], of which we retain 47 hours for training and 18 hours for test. Videos were manually annotated for speech, music, noise, and singing segments by annotators at the Linguistic Data Consortium using the XTrans tool [12] and targeting a "quick rich transcription". The music, noise, and singing segmentatins were used to defined 4 (not mutually exclusive) environments in which the speech/non-speech decision is made: music present, noise present, singing present, and clean. The prevalence of these different environment types in the corpus is depicted in Table 1.

		Environment			
	Overall	Music	Noise	Singing	Clean
Train	46.78	21.86	16.23	9.71	11.34
Test	17.66	8.52	6.20	3.45	4.02

Table 1: Total amount of audio (hours) in corpus containing each environment.

2.2. Evaluation metrics

We evaluate the frame-wise performance of the systems using four error metrics:

1. *Error rate (ER)* The percentage of misclassified frames.
2. *Miss rate (MR)* The percentage of speech frames classified as non-speech.
3. *False alarm rate (FAR)* The percentage of non-speech frames classified as speech.
4. *Equal error rate (EER)* The error rate at the point on the Receiver Operating Characteristic (ROC) curve at which MR and FAR are equal.

Note that no forgiveness collar [13] is used during the evaluations.

3. GMM Baseline System

3.1. Features

We extract 13 MFCC features every 10 ms using a 25 ms analysis window. These features are then normalized on a file-by-file basis to have zero mean and identity covariance. The normalized features, along with their first and second differences, are concatenated to form a 39-dimentional feature vector for input to the baseline GMM system.

3.2. Model

As our baseline we consider a GMM classifier with two classes: speech and non-speech. Each class was modeled by a GMM with 128 components and diagonal covariance matrix structure. GMMs were initialized by performing one round of K-Means clustering using the entire training set, then refined by running 20 iterations of the Expectation-Maximization (EM) algorithm.

3.3. Segmentation

We consider two segmentation schemes. The first makes frame-wise speech/non-speech decisions by comparison of the log-likelihood ratio (LLR) between speech and non-speech GMMs to a threshold, as in [14]. Obviously, the performance of the system is heavily dependent on the chosen threshold, with different values being optimal for different contexts. For ease of reporting, we follow [6, 9] and choose the threshold so that false alarm and miss rates are equal on the test set and report the corresponding EER.

In our second segmentation scheme, we produce frame-wise decisions by Viterbi decoding of the GMM log-likelihoods with a 2-state (speech/non-speech) HMM. The HMM state-transition probabilities and state priors are set to the empirically observed values in the training set.

3.4. Performance

In Table 2 we present overall EER of the 128-component GMM baseline system along with EER within each environment type. Additionally, as an exploration of the importance of the exact number of components used in training the system, we compare to systems with 64 and 512 components. While there is a benefit in using more components, this seems to fall off above 128 mixture components with a 12.26% reduction in overall EER going from 64 to 128 components, but $< 0.01\%$ reduction in overall ERR going from 128 to 512 components. Across environment types, EERs are highest for segments containing music or singing and lowest for clean or noisy conditions.

Post-Viterbi decoding frame-wise ERs, MRs, and FARs for the 128-component GMM baseline are presented in Table 3. Here, we see that use of Viterbi decoding results in a 8.41% (relative) reduction in overall frame-wise error rate. Overall and within each environment individually, false alarms appear less prevalent that misses.

	Environment				
#Components	Overall	Music	Noise	Singing	Clean
64	45.56	45.71	46.19	49.65	42.02
128	39.97	43.64	38.91	47.55	33.87
512	39.94	43.81	38.52	47.86	32.64

Table 2: EER (%) in different environments for GMM baseline system (128 components) and systems trained with 64 and 512 components.

	Environment				
	Overall	Music	Noise	Singing	Clean
ER	36.61	37.64	34.29	40.0	37.43
FAR	31.06	35.33	25.25	38.27	24.57
MR	47.59	50.62	52.35	55.25	43.17

Table 3: ER, FAR, and MR (%) for 128 component GMM baseline system after Viterbi decoding.

4. Deep Neural Network

4.1. Features

As with the GMM baseline system, we extract 13 MFCC features every 10 ms using a 25 ms analysis window, which are then normalized on a file-by-file basis to have zero mean and identity covariance. At each frame we concatenate the 13 normalized MFCCs extracted at that frame with those of the 40 immediately preceding and following frames (an 81 frame context window), yielding a 1,053 component feature vector that serves as the input to our deep neural network.

4.2. Model

We train a deep neural network [10] with the following architecture: a 1,053 unit input layer, 3 hidden layers, each containing 512 Rectified Linear Units [15], and an output layer consisting of two softmax units. The network was trained by backpropagation for 50 epochs (an epoch consisting of 100,000 examples) using mini-batch gradient descent with a mini-batch size of 50 and learning rate of 0.001. Training was accelerated by use of a momentum of 0.9. No pretraining was performed[1].

4.3. Segmentation

As with the GMM baseline system, we consider two segmentation schemes. The first makes frame-wise speech/non-speech decisions by thresholding on the posterior probability of speech being present that is produced by the DNN. As before, we choose this threshold so that false alarm and miss rates are equal on the test set and report the corresponding EER.

In our second segmentation scheme we produce frame-wise decisions by Viterbi decoding using a 2-state (speech/non-speech) HMM whose state-transition probabilities and state priors are set at the observed values in the training set. As input to the Viterbi decoder we use scaled speech/non-speech log-likelihoods, produced by dividing the state posteriors (as estimated by the DNN) by the state priors (as estimated on the training set) [16].

4.4. Performance

Frame-wise EERs (overall and for each environment seperately) for the DNN SAD system are reported in Table 4. Relative to the GMM baseline, we observe a 50.86% reduction in overall EER. Similarly large reductions in ERR are observed for each environment individually, though performance is noticeably worse when singing is present (33.52%) compared to all other conditions. The results of Viterbi decoding of the DNN outputs are given in Table 5. As was observed with the GMM baseline,

[1]While it is typical practice to generatively pretrain a DNN using stacked Restricted Boltzman Machines or Denoising Autoencoders, we have found that for Speech Activity Detection, such pretraining has little benefit when using Rectified Linear Units; at least, for the network depths considered in this paper.

Environment				
Overall	Music	Noise	Singing	Clean
19.64	23.36	22.14	33.52	19.82

Table 4: EER (%) in different environments for DNN system.

Viterbi decoding results in a marked decrease in overall frame-wise error rate: 19.64% to 16.61% or a 15.43% reduction. Similarly large improvements are seen for SAD in segments containing music, noise, or singing but not, strikingly, under clean conditions: in fact, post-Viterbi decoding ER under clean conditions is nearly double that for environments containing music and 19.08% higher than for noisy environments. We believe that this is due to two factors: first, the system correctly recognized as non-speech many fairly long pauses (up to 800 msec) that the Gold Standard annotation treated as part of speech regions; and second, speech segments were a relatively small proportion of the overall training materials (24.23%), and so inclusion of the prior probabilities tended to push marginal cases (of which there are plenty) into the non-speech category.

	Environment				
	Overall	Music	Noise	Singing	Clean
ER	16.61	11.45	18.83	14.58	23.27
FAR	6.68	5.40	6.73	9.08	12.50
MR	36.24	45.37	42.72	61.20	28.08

Table 5: ER, FAR, and MR (%) for the DNN system after Viterbi decoding.

4.5. Why does the DNN system outperform the GMM baseline?

Given the results achieved above with the DNN system, one might ask where this impressive improvement in performance is coming from. One possibility is that these performance gains are entirely due to the additional information present in the extended context window features used by the DNN. Undoubtledy, some of the performance improvement is due to these extended context features, but it may also be the case that DNNs are more powerful than diagonal covariance GMMs even when the feature sets are identical. Ideally, then, we would compare the performance of the two architectures using a fixed feature set.

As diagonal covariance matrix GMMs are ill-suited to handling highly correlated features, we were unable to directly compare the systems using the full 81-frame context window feature set of Section 4.1. Consequently, here we train a DNN system as in Section 4.2, but using the same feature set used in GMM training (the 13 normalized MFCC features with their deltas and delta-deltas, as described in Section 3.1). As seen in Tables 6 and 7, even when using the same features the DNN based system outperforms the GMM baseline, suggesting that the DNN has superior modeling ability to diagonal covariance matrix GMMs.

Environment				
Overall	Music	Noise	Singing	Clean
31.07	33.35	34.80	39.53	31.85

Table 6: EER (%) in different environments for DNN system trained using GMM baseline features.

	Environment				
	Overall	Music	Noise	Singing	Clean
ER	25.43	15.54	26.68	15.93	42.78
FAR	5.44	5.72	4.96	8.48	5.07
MR	64.95	70.54	69.56	79.04	59.61

Table 7: ER, FAR, and MR (%) after Viterbi decoding for the DNN system trained using GMM baseline features.

4.6. Context window size

While the above suggests that some of the DNN based system's performance improvement is due to simply being more efficient at modeling, it is also clear that the system is benefitting from the additional information in the extended context window it uses as input. A natural question that arises, then, is just how large a context window is necessary for accurate SAD? Consequently, we trained a series of DNN based SAD systems of the same form as that presented in Section 4.2, but with different context window lengths (*nf*, the number of frames in the context, in {1, 11, 21, 32, 41, 51, 61, 71, 81}). In Figure 1 we plot both EER and post-Viterbi decoding ER as a function of context window length. There is a definite benefit to incorporating longer context windows, though most of the improvement can be achieved using a context of only 21 frames, as compared to the full 81 frame context.

5. Discussion

Speech Activity Detection of web videos is difficult for human listeners as well as for machines. The sound tracks of user-generated web videos sometimes involve indistinct speech whose boundaries and even existence are uncertain; the distinction between speech and various sorts of non-speech human-created sounds is often fuzzy; even in a stream of relatively clear speech, there are uncertainties about how to treat pauses. No estimates of inter-annotator agreement are available for the HAVIC data, so we created our own estimate by selecting thirty 10-second segments at random from the HAVIC test data, and annotating these ourselves for speech vs. non-speech. The result disagreed with the "gold standard" for the same segments on 13.6% of all frames, suggesting that there is probably an effective noise floor of 10-15% frame-wise overall ER in this material, in the sense that independent human annotators will disagree with one another at about that rate. In this context the 16.61% frame-wise overall ER achieved by our best system is certainly respectable. Moreover, it compares favorably to 19.6% reported for similar materials by [9], though obviously differences in the make up of the test sets and annotation guidelines make direct comparison impossible.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0964556.

7. References

[1] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke, "The Meeting Project at ICSI," in *Proceedings of the First International Conference on Human Language Technology Research*, 2001, pp. 1–7.

[2] T. Pfau, D. P. Ellis, and A. Stolcke, "Multispeaker Speech Activity Detection for the ICSI Meeting Recorder," in *Proceedings*

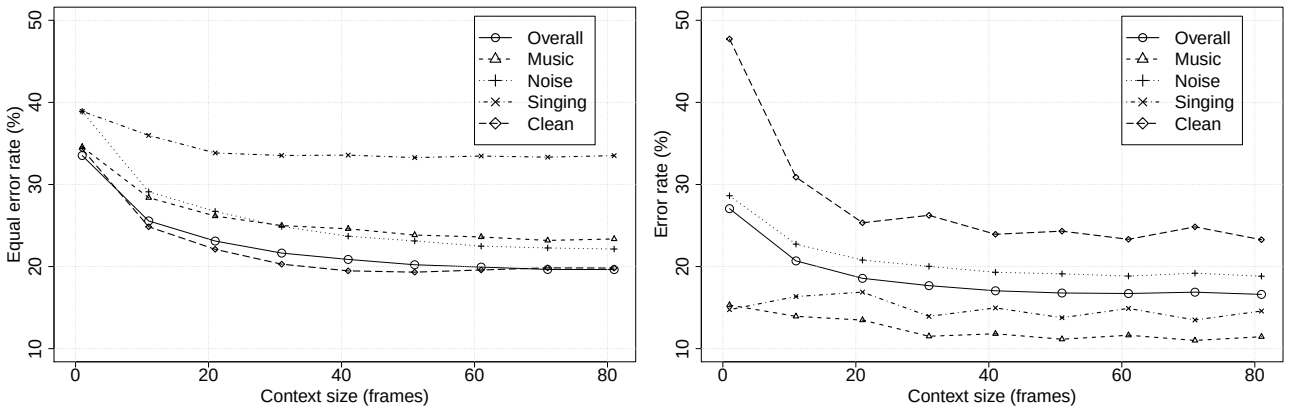

Figure 1: Frame-wise results for DNN sytems trained with differing context window sizes. *Left:* Equal error rates (%). *Right:* Error rates (%) after Viterbi decoding.

of Automatic Speech Recognition and Understanding, 2001, pp. 107–110.

[3] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *Proceedings of ICASSP*, vol. 4, 2002, pp. 4072–4075.

[4] S. Sadjadi and J. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *Signal Processing Letters, IEEE*, vol. 20, pp. 197–200, 2013.

[5] N. Mesgarani, M. Slaney, and S. A. Shamma, "Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 920–930, 2006.

[6] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, K. Vesely, P. Matejka, X. Zhu, and N. Mesgarani, "Developing a speech activity detection system for the DARPA RATS program," in *Proceedings of InterSpeech*, 2012.

[7] A. Saito, Y. Nankaku, A. Lee, and K. Tokuda, "Voice activity detection based on conditional random fields using multiple features," in *Proceedings of InterSpeech*, 2010, pp. 2086–2089.

[8] P. Clement, T. Bazillon, and C. Fredouille, "Speaker diarization of heterogeneous web video files: A preliminary study," in *Proceedings of ICASSP*, 2011, pp. 4432–4435.

[9] A. Misra, "Speech/nonspeech segmentation in web videos," in *Proceedings of InterSpeech*, 2012.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] S. Strassel, A. Morris, J. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, and M. Michel, "Creating HAVIC: Heterogeneous Audio Visual Internet Collection," in *Proceedings of the Eight International Conference on Language Resources and Evaluation*, 2012.

[12] M. L. Glenn, S. M. Strassel, and H. Lee, "XTrans: A speech annotation and transcription tool," *Proceedings of InterSpeech*, 2009.

[13] J. Fiscus, N. Radde, J. Garofolo, A. Le, J. Ajot, and C. Laprun, "The Rich Transcription 2005 Spring Meeting Recognition Evaluation," *Machine Learning for Multimodal Interaction*, pp. 369–389, 2006.

[14] J. Ramírez, J. C. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *Signal Processing Letters, IEEE*, vol. 12, no. 10, pp. 689–692, 2005.

[15] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th International Conference on Machine Learning*, 2010, pp. 807–814.

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