The IBM Speech Activity Detection System for the DARPA RATS Program

George Saon, Samuel Thomas, Hagen Soltau, Sriram Ganapathy and Brian Kingsbury

IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
e-mail: gsaon@us.ibm.com

Abstract

We present the IBM speech activity detection system that was fielded in the phase 2 evaluation of the DARPA RATS (robust automatic transcription of speech) program. Key ingredients of the system are: multi-pass HMM Viterbi segmentation, fusion of multiple feature streams, file-based and speech-based normalization schemes, the use of regular and convolutional deep neural networks, and model fusion through frame-level score combination of channel-dependent models. Using these techniques, our system achieved an excellent performance during the RATS phase 2 evaluation.

Index Terms: speech activity detection, robust speech recognition

1. Introduction

The goal of the DARPA RATS program is to develop techniques for performing speech activity detection (SAD), language identification (LID), speaker identification (SID) and keyword search (KWS) in multiple languages on degraded audio signals transmitted over communication channels that are extremely noisy and/or highly distorted. The speech activity detection task deals with determining whether a signal contains speech or is just comprised of background noise or music. The segmented speech regions can be send downstream to the other components (LID, SID and KWS) for further processing as done in [1] or can be directly used by analysts. Given its importance in the context of this program, SAD is evaluated in isolation of the other components. The performance metric used in this paper is the equal error rate which is defined as the point where the probability of miss (P_{Miss}) coincides with the probability of false accept (P_{FA}). These two quantities are defined as the duration of missed speech over the duration of total speech and the duration of false accept speech over the duration of total non-speech, respectively.

The paper is organized as follows: in section 2 we describe the system architecture, feature extraction, normalization and segmentation models; in section 3 we present some experimental results, and in section 4 we summarize our findings and propose future directions.

2. System overview

The operation of our system may be broken down into three stages depicted in Figure 1: (1) channel detection with 8 channel-dependent Gaussian mixture models trained with maximum likelihood on a fusion of PLP and voicing features (see 2.3.1), (2) speech/non-speech HMM Viterbi segmentation using channel-dependent deep neural networks (DNNs) trained on a fusion of PLP, voicing and rate-scale features with file-based mean and variance normalization (see 2.3.3) and (3) speech/non-speech HMM Viterbi segmentation using a frame-level score combination of three sets of channel-dependent neural networks: (i) the models from (2), (ii) DNNs trained on a fusion of PLP, voicing and FDLP features with speech-based mean and variance normalization (see 2.3.2) and (iii) deep convolutional neural nets (CNNs) trained on log-mel spectra with speech-based mean and variance normalization (see 2.3.4). The speech segments needed for speech-based normalization are hypothesised in pass (2).

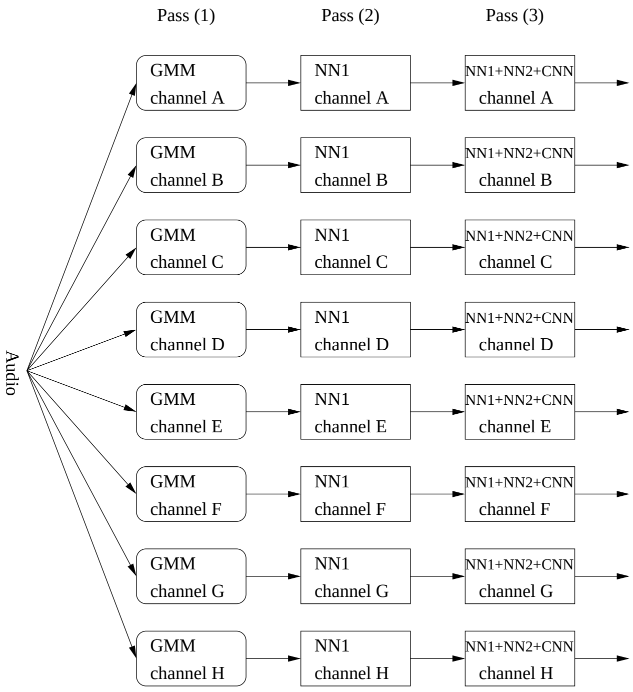

Figure 1: System diagram.

2.1. HMM Viterbi segmentation

We propose to treat the segmentation problem as a simple ASR decoding problem with a three word vocabulary (S, NS, NT) similar to [2]. The HMM topology used for Viterbi decoding is shown in Figure 2. All 5 states for a given “word” share the same output distribution. Analogous to the LM score, the segment insertion penalty controls the number (and duration) of the segments. The tradeoff between missed speech and inserted speech is controled by adding a fixed threshold to the S scores for every frame. The frame-level scores are scaled by an acoustic weight of 0.03 for all the experiments. Following the decoding, the boundaries of the hypothesized speech segments are extended by an additional 0.1 seconds to capture low energy speech as suggested in [3].

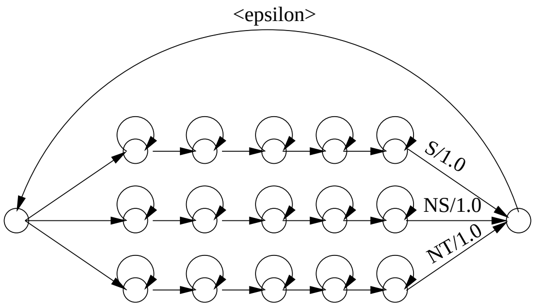

Figure 2: HMM topology for Viterbi segmentation.

2.2. Training and test data

The audio data consists of recordings from existing speech corpora such as the Fisher English and Arabic Levantine conversational telephone speech as well as new recordings that were specifically collected for the RATS program (telephone conversations in Arabic Levantine, Pashto and Urdu). These recordings were transmitted through 8 different communication channels denoted by the letters A through H. For the purpose of SAD, the annotations consist of time-marked regions that are labeled either as speech (S), non-speech (NS) or non-transmission (NT). The annotations were created by mapping the labels and time marks obtained by an automatic segmentation of the original (i.e. non-retransmitted) files. Because of this automatic annotation scheme, there is a staggering amount of training and test data available. The training data consists of 2034 hours of audio and was distributed to the RATS participants by the Linguistic Data Consortium (LDC) in three incremental releases. For faster experimental turnaround, we also subsampled the training data at the segment level by a factor of 10. We report results on the official DEV1 and DEV2 testsets which contain 11 hours and 20 hours of audio, respectively.

2.3. Feature extraction

2.3.1. PLP and voicing features

The first feature set consists of 13-dimensional PLP cepstra extracted every 10ms within a 25ms sliding window. Similar to [3], the cepstra are normalized to zero mean and unit variance using either audio file-based statistics or speech-only based statistics depending on the models. Additionally, we apply ARMA-filtering [4] for each dimension within a temporal window of ±20 frames. We found this to be slightly better than likelihood averaging as proposed in [3]. To each normalized PLP frame we append a 1-dimensional probability of voicing feature [5] yielding a 14-dimensional frame. Every 17 consecutive PLP+voicing frames are spliced together and projected down to 40 dimensions using linear discriminant analysis (LDA). As noted in [3], standard LDA for a three class (S,NS,NT) problem can only find two dimensions because of the rank of the between-class covariance matrix. Our workaround was to use a Gaussian-level LDA where we train 32 Gaussians per class and declare the Gaussians as LDA classes. This has the effect of splitting each class into several subclasses and results in more accurate decision boundary modeling.

2.3.2. FDLP features

Frequency domain linear prediction (FDLP) is a technique for autoregressive modeling of the Hilbert envelopes of the signal [6]. This is achieved by the application of linear prediction on the discrete cosine transform (DCT). The FDLP technique is used for feature extraction of speech by windowing the DCT of a long-term segment (1000 ms). This is followed by the linear prediction of sub-band DCT components to yield temporal envelopes in each band [7]. The sub-band envelopes are integrated in short-term windows (25 ms with a shift of 10 ms) to derive a spectrographic representation of the speech signal and the cepstral transformation is applied to derive 13 dimensional features. Similar to PLP processing, the FDLP cepstra are normalized to zero mean and unit variance using speech-only based statistics and the same ARMA filtering is applied. FDLP frames within a ±8 frames context window are spliced together and projected down to 40 dimensions by means of a Gaussian-level LDA transform.

2.3.3. Rate-scale features

Inspired by [3], a feature extraction technique is developed based on spectro-temporal modulation filtering of the auditory spectrogram [8]. The two dimensional spectrographic representations are derived by emulating various processing stages in the periphery of the human auditory system. The auditory spectrogram is then transformed to modulation domain using Fourier transforms along the spectral and temporal axis and modulation filtering is applied to extract key dynamics in the scale and rate dimensions respectively [9]. The modulation filters used in this feature extraction scheme are broad enough to cover a wide range of dynamics (0-2 cycles per octave in the scale dimension and 0.25-25 Hz in the rate dimension). Cepstral transformation is applied on the filtered auditory spectrograms to obtain 13 dimensional features. Similar to PLP processing, the rate-scale cepstra are normalized to zero mean and unit variance using audio file-based statistics and the same ARMA filtering is applied. Rate-scale frames within a ±8 frames context window are concatenated and projected down to 40 dimensions by means of a Gaussian-level LDA transform.

2.3.4. Log-mel spectral features

We also extracted log-mel spectra which have the property that neighboring dimensions in time and frequency are highly correlated. This locality property is important for training convolutional neural nets. We opted for a 40-dimensional Mel filterbank spanning the entire 0-8Khz frequency range. The log-energies are normalized to zero mean and unit variance using speech-only based statistics.

2.4. Acoustic modeling

2.4.1. GMMs for channel detection

Channel information is not provided to the SAD system during testing and must be inferred. Note that the channel set is closed i.e. the same eight channels are present during training and testing. Channel detection is necessary in order to use channel-dependent modeling. Our approach to channel detection was to train 8 channel-dependent GMMs with 3072 diagonal covariance Gaussians each. The Gaussians were estimated using maximum likelihood on 40-dimensional PLP and voicing features described in 2.3.1. All Gaussians are scored for every frame and the GMM with the highest total likelihood determines the channel. This approach has 100% channel detection accuracy on the two official testsets (DEV1 and DEV2).

Model	DEV1	DEV2
GMM	1.99	3.26
NN 1 hidden	1.79	2.59
NN 2 hidden	1.68	2.62

Table 1: Equal error rates (%) on DEV1 and DEV2 for GMMs and neural networks trained on 1/10th of the data in the same feature space (PLP+voicing).

2.4.2. *Deep neural networks*

Inspired by the recent success of deep neural networks for general ASR [10], we experimented with such models for segmentation. The methodology for training the nets is as follows. We fully train a network with N hidden layers (with 1024 units) which is then used to initialize a network with $N + 1$ hidden layers which in turn is also fully trained. The training step size is annealed (i.e. halved) whenever the cross-entropy criterion decreases on some held-out data (typically 1/10th of the training data). It usually takes around 30 passes (epochs) through the training data for the training to converge.

2.4.3. *Convolutional neural networks*

We also experimented with convolutional neural networks (CNNs) [11] which are designed to handle distortions in the frequency domain. The network structure is as follows. The input features are 40-dimensional log-mel spectra (see 2.3.4) augmented with first and second order derivatives resulting in 3 blocks. 3 sliding windows of size 9×9 cover an input context of 11 frames for each block. At the frequency level, we get $40 - 9 + 1 = 32$ positions. Accounting for the Δ and $\Delta\Delta$ features, we get $3 \times 32 = 96$ windows, and each window has $3 \times 9 \times 9 = 243$ features. The first hidden layer performs maximum pooling in frequency. The second hidden layer is also convolutional and uses a sliding window over the previous layer. More details about this architecture and training can be found in [12].

3. Experiments and results

In this section we describe the various experiments that were performed and their impact on the segmentation performance. More concretely, we discuss the benefit of using neural networks over GMMs, the effect of feature fusion, the effect of using more training data, and the effect of model combination on system performance.

3.1. Neural networks versus GMMs

In this experiment, we compare channel-dependent GMMs with 1024 40-dimensional Gaussians/class with channel-dependent neural networks with one and two hidden layers. Both sets of models are trained on 1/10th of the data on 40-dimensional PLP and voicing features as described in 2.3.1. The neural networks use 40-dimensional features augmented with single, double and triple deltas leading to an input layer of size $40 \times 4 = 160$. The hidden layers have 1024 neurons each and the output layer has 3 neurons corresponding to S, NS and NT.

As can be seen from Table 1, the use of neural networks leads to a 15% relative improvement in equal error rate over GMMs. In light of this result, GMMs were abandoned in favor of neural networks in all subsequent experiments.

Model	CMVN	DEV1	DEV2
PLP+v	file-based	1.68	2.62
PLP+v+FDLP	file-based	1.52	2.25
PLP+v+FDLP	speech-based	1.36	2.24
PLP+v+RS	file-based	1.51	2.26
PLP+v+RS	speech-based	1.37	2.27

Table 2: Equal error rates (%) on DEV1 and DEV2 for neural networks trained on 1/10th of the data on various feature streams with either file-based or speech-based normalization (RS stands for rate-scale features 2.3.3).

Model	Data	DEV1	DEV2
PLP+v+FDLP-F	1/10th	1.52	2.25
PLP+v+FDLP-F	all	1.16	2.06
PLP+v+FDLP-S	1/10th	1.36	2.24
PLP+v+FDLP-S	all	1.01	1.96
PLP+v+RS-F	1/10th	1.51	2.26
PLP+v+RS-F	all	1.14	2.01

Table 3: Equal error rates (%) on DEV1 and DEV2 for neural networks trained on subsampled and entire data on various feature streams (-F,-S stand for file-based and speech-based normalization, respectively).

3.2. Feature fusion

Feature fusion is a powerful technique for speech activity detection as shown in [3, 13]. We train neural networks on fused feature streams obtained by adding various 40-dimensional features to the 40-dimensional PLP+voicing stream. As before, the resulting 80-dimensional vector is augmented by finite differences up to order 3 yielding an input of size $80 \times 4 = 320$. The idea in this set of experiments is to let the neural network perform an optimal feature stream combination as opposed to using ad-hoc feature stream weights. All networks in these experiments have two hidden layers with 1024 hidden units.

We observe from Table 2, that adding either the FDLP or the rate-scale feature stream results in an additional 10% improvement in the segmentation performance. There is no additional gain when adding both FDLP and RS streams to the PLP+voicing features. Also, models trained with speech-based cepstral mean and variance normalization are more effective on DEV1 than models using file-based CMVN.

3.3. Amount of training data

In this subsection, we compare training the neural networks on the subsampled data versus training them on the entire data. The best configuration when training on all the data resulted in models with 3 hidden layers (as opposed to 2 layers for the ones estimated on the subsampled data). Surprisingly, even though SAD is deemed to be a “simple” problem, training on 2000 hours of audio helps by an additional 20% relative for all the models as can be seen from Table 3. This is in contrast to GMM training where, in some earlier experiments, adding more data on top of the subsampled data did not help the segmentation performance.

3.4. Model fusion

Finally, we experimented with model fusion using a weighted log-linear frame-level score combination of three sets of channel dependent networks at the frame level. The three models that were combined are: (i) nets trained on a fusion of PLP, voicing and rate scale features with file-based normalization (the models from pass (2)), (ii) nets trained on a fusion of PLP, voicing and FDLP features with speech-based normalization, and (iii) convolutional nets trained on log mel spectral features with speech-based normalization. These models were trained on all the data and were selected for diversity and because of similar performance which is desirable for system combination. Model fusion was also preferred over additional feature fusion because adding more feature streams results in prohibitive disk space and training time requirements when training on all the available data. In Figure 3 we show the ROC curves for the individual networks and the model fusion on DEV1 and DEV2.

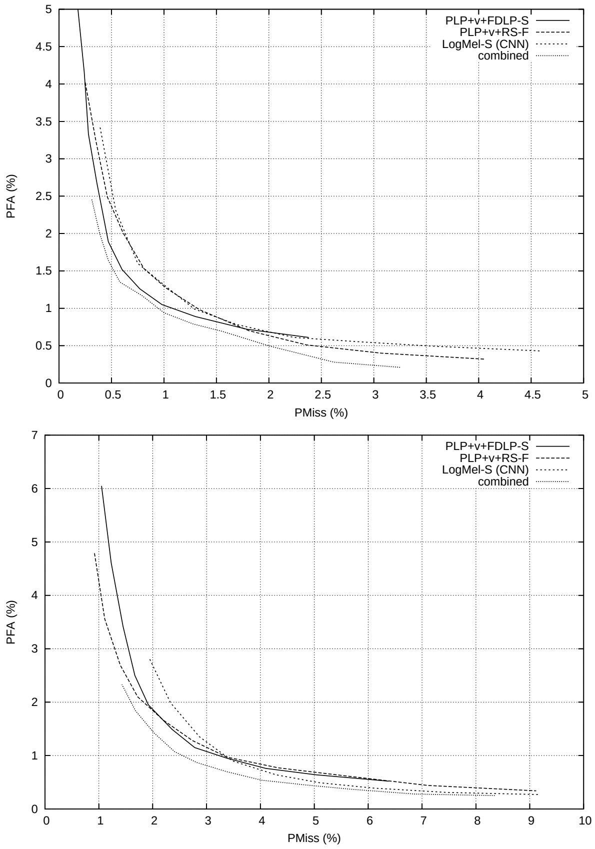

Figure 3: ROC curves for individual networks and model fusion on DEV1 (top) and DEV2 (bottom).

4. Conclusion

We have presented the SAD system developed by IBM for the RATS phase 2 evaluation. The system achieves equal error rates between 1% and 1.7% depending on the testset and has shown excellent results on the unseen evaluation data. The techniques that were instrumental in reaching this level of performance are: the use of regular and convolutional channel-dependent neural networks, combining multiple feature streams that differ in type and normalization (file-based and speech-based CMVN), training on all of the available data, and model fusion by combining the frame-level scores of neural networks that differ in input features and type (regular and convolutional). Future work will address supervised and unsupervised adaptation to unseen channels.

5. Acknowledgments

The authors wish to thank Tomas Beran for help with software optimization. This work was supported in part by DARPA under Grant HR0011-12-C-0015[1].

6. References

[1] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, “Exploiting diversity for spoken term detection,” in *Proc. of ICASSP*, 2013.

[2] G. Saon, G. Zweig, B. Kingsbury, L. Mangu, and U. Chaudhari, “An architecture for rapid decoding of large vocabulary conversational speech,” in *Proc. Eurospeech*, 2003.

[3] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarami, K. Vesely, and P. Matejka, “Developing a speech activity detection system for the DARPA RATS program,” in *Proc. Interspeech*, 2012.

[4] C.-P. Chen and J.A. Bilmes, “MVA processing of speech features,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 1, 2007.

[5] A. de Cheveigne and H. Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *Journal of the Acoustical Society of America*, vol. 111, no. 4, 2002.

[6] M. Athineos and D. Ellis, “Autoregressive modelling of temporal envelopes,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, 2007.

[7] S. Ganapathy, *Signal Analysis using Autoregressive models of amplitude modulaton*, Ph.D. thesis, Johns Hopkins University, 2012.

[8] T. Chi, P. Ru, and S.A. Shamma, “Multiresolution spectrotemporal analysis of complex sounds,” *Journal of the Acoustical Society of America*, vol. 118, 2005.

[9] S. Nemala, K. Patil, and M. Elhilali, “A multistream feature framework based on bandpass modulation filtering for robust speech recognition,” *IEEE Trans. on Audio, Speech and Language Processing*, vol. 21, no. 2, 2013.

[10] F. Seide, G. Li, X. Chen, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proc. Interspeech*, 2011.

[11] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Proc. NIPS*, 1990.

[12] H. Soltau et al., “Acoustic modeling for the DARPA RATS program,” in *Proc. Interspeech*, 2013.

[13] S. Sadjadi and J. Hansen, “Unsupervised speech activity detection using voicing measures and perceptual spectral flux,” *IEEE Signal Processing Letter*, vol. 20, no. 3, 2013.

[1] Approved for Public Release, Distribution Unlimited. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.