

The IBM RT06s Evaluation System for Speech Activity Detection in CHIL Seminars

Etienne Marcheret, Gerasimos Potamianos,
Karthik Visweswariah, and Jing Huang

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
{etiennem,gpotam,kv1,jhgh}@us.ibm.com

Abstract. In this paper, we describe the IBM system submitted to the NIST Rich Transcription Spring 2006 (RT06s) evaluation campaign for automatic speech activity detection (SAD). This SAD system has been developed and evaluated on CHIL lecture meeting data using far-field microphone sensors, namely a single distant microphone (SDM) configuration and a multiple distant microphone (MDM) condition. The IBM SAD system employs a three-class statistical classifier, trained on features that augment traditional signal energy ones with features that are based on acoustic phonetic likelihoods. The latter are obtained using a large speaker-independent acoustic model trained on meeting data. In the detection stage, after feature extraction and classification, the resulting sequence of classified states is further collapsed into segments belonging to only two classes, speech or silence, following two levels of smoothing. In the MDM condition, the process is repeated for every available microphone channel, and the outputs are combined based on a simple majority voting rule, biased towards speech. The system performed well at the RT06s evaluation campaign, resulting to 8.62% and 5.01% “speaker diarization error” in the SDM and MDM conditions respectively.

1 Introduction

Speech activity detection (SAD) has long been an important issue as a front end step to the *automatic speech recognition* (ASR) process. Its significance ranges (although not limited to) from bandwidth usage in the client/server ASR paradigm, to stable prompt control during barge-in operation. SAD has a positive impact on ASR in terms of both CPU usage and accuracy, since the decoder is not required to operate on non-speech segments, reducing processing effort and word insertion error rate. Furthermore, robust performance of SAD is crucial in developing technologies for the smart room domain, for example lecture seminars and meetings, as in the CHIL (“Computers in the Human Interaction Loop”) project [1]. There, in addition to ASR, SAD is useful as a pre-processing step for speaker localization.

Not surprisingly, speech activity detection has attracted significant interest in the ASR literature. Most techniques are based on features extracted from the acoustic signal, ranging from energy [2] to frequency-based representations of speech [3, 4, 5]. The selected features are subsequently used in speech/ silence classification, ranging from adaptive thresholding to linear discriminants,

regression trees, distance measures, and Gaussian mixture model (GMM) based classifiers. In general, energy-based SAD is computationally efficient and simple to implement, but it lacks robustness to noise. Performance can be improved by using adaptive thresholds or appropriate filtering of the energy estimates [2, 6], however addressing non-stationary noise effectively remains difficult. Most often, frequency-based speech features, such as mel-frequency cepstral coefficients (MFCCs), are required to achieve improved robustness to noise.

In previous work [7, 8], we have proposed to employ such MFCC features indirectly, through the acoustic model that is assumed to generate them. The resulting acoustic phonetic features were extracted based on the phonetic class conditional observation vector likelihoods by the acoustic model, and were used to augment traditional energy-like based features. The two types of features were subsequently fused with a simple concatenation and a projection-based dimensionality reduction, and fed to a Gaussian mixture classifier for speech/silence detection. Among other domains, the proposed algorithm was tested on single-microphone, far-field acoustic data, collected as part of the CHIL project [8], during the first CHIL-internal evaluation campaign (“CHIL eval. run #1”), achieving excellent results.

Since then, a number of slight modifications have been applied to the IBM SAD system in an effort to improve its performance for the “Rich Transcription Spring 2006” (RT06s) evaluation campaign. In particular, the diagonal covariance GMM classifier was replaced by a full covariance model operating on a reduced set of features, by eliminating highly correlated features. In addition, the acoustic model has been replaced by a PLP-feature based model developed for the RT06s speech-to-text evaluation [9]. Finally, to allow operation on multiple microphone inputs, a simple majority voting scheme has been implemented to combine single-microphone SAD system outputs. This corresponds to channel integration (fusion) in the “decision” level, instead of the “signal” level followed by other works in the literature [10, 11].

Our SAD system improvements are discussed in detail in Sections 2, 3, and 4. In particular, Section 2 is devoted to feature extraction, Section 3 to SAD system training, and Section 4 focuses on SAD testing. A number of development experiments and SAD evaluation results are presented in Section 5. Finally, Section 6 concludes the paper with a short summary.

2 Feature Extraction for SAD

As discussed in the Introduction, the IBM SAD system operates on two types of features: Energy based ones, generated directly from the waveform, and acoustic phonetic features, defined from observations generated by the ASR acoustic model. The two feature sets are combined, and are subsequently fed to a Gaussian mixture model (GMM) classifier, as discussed in Section 3.

2.1 Energy Based Features

The energy based feature space is defined by a five-dimensional vector, the components of which are based on the bandpass filtered acoustic waveform within

the [200, 900] Hz range. Letting $y[i]$ denote the bandpass-filtered waveform at sample time i , the estimated short time energy $e(t)$ for a window of length N is given by

$$e(t) = 10 \log \left(\frac{1}{N} \sum_{i=1}^N y[i]^2 \right), \quad (1)$$

measured in dB. In (1), t is discrete and determined by the observation frame rate, set in this work to be every 10 ms. This results to $N = 160$ samples in (1) for 16 kHz audio. Given $e(t)$, we generate filtered observations of it, based on

$$rms(t) = 10^{scale \times e(t)}. \quad (2)$$

In (2), $rms(t)$ is defined as a linear energy, scaled for the expected number of bits of resolution. The scaling constant is given by $scale = contrast/scaleMax$, where the *contrast* provides a level of sensitivity, generally set within [3.5, 4.5], and *scaleMax* is the maximum possible value of $e(t)$, for example 90.3 dB for a 16-bit signed linear PCM signal.

Based on the instantaneous $rms(t)$ value, we can obtain “low”, “mid”, and “high” energy tracks, defined as

$$lt(t) = (1 - \alpha_{l,t}) \times lt(t-1) + \alpha_{l,t} \times rms(t) \quad (3a)$$

$$mt(t) = (1 - \alpha_m) \times mt(t-1) + \alpha_m \times rms(t) \quad (3b)$$

$$ht(t) = (1 - \alpha_{h,t}) \times ht(t-1) + \alpha_{h,t} \times rms(t) \quad (3c)$$

respectively. For the mid-track $mt(t)$, time constant α_m is fixed, set in this work to 0.1. Therefore $mt(t)$ is the lowpass filtered $rms(t)$. The remaining low and high track time constants $\alpha_{l,t}$ and $\alpha_{h,t}$ are functions of the instantaneous $rms(t)$, designed so that rapid changes in energy will cause abrupt tracking by $lt(t)$ and $ht(t)$, respectively. They are given by

$$\alpha_{l,t} = \left(\frac{lt(t-1)}{rms(t)} \right)^2 \quad \text{and} \quad \alpha_{h,t} = \left(\frac{rms(t)}{ht(t-1)} \right)^2,$$

thus resulting in increasing $\alpha_{l,t}$ for decreasing $rms(t)$, and increasing $\alpha_{h,t}$ for increasing $rms(t)$.

Next, from (3), we form three equivalent low, mid, and high energy representations, given by

$$let(t) = \frac{\log(lt(t))}{scale}, \quad met(t) = \frac{\log(mt(t))}{scale}, \quad \text{and} \quad het(t) = \frac{\log(ht(t))}{scale}. \quad (4)$$

From (4), we can also obtain the mid-to-low energy track relationship as

$$m2l(t) = met(t) - let(t). \quad (5)$$

By combining (1), (4), and (5) we obtain a five-dimensional energy feature vector at frame t , as

$$v_e(t) = [e(t) \ let(t) \ met(t) \ het(t) \ m2l(t)]. \quad (6)$$

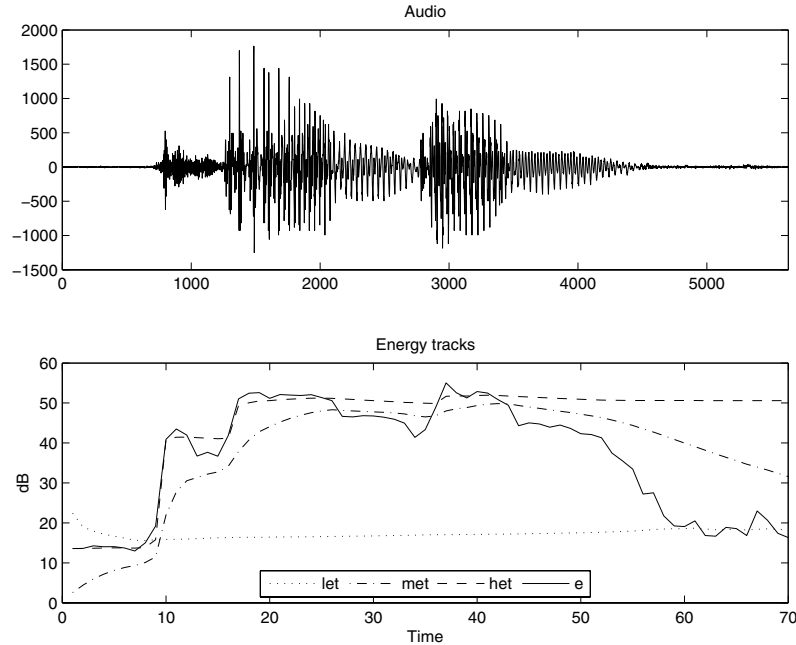


Fig. 1. Audio waveform and corresponding energy tracks

A purely energy based speech activity detector based on these observations where dynamic speech/silence thresholds are employed can be found in [6].

Fig. 1 shows the bandpass filtered energy (1) and the corresponding energy tracks *let*, *met*, and *het* defined in (4). We observe that the low and high energy tracks are intended to lock onto the “floor” and speech signal levels respectively, while the mid track is a lowpass filtered energy track.

2.2 Acoustic Phonetic Features

The acoustic phonetic feature space employed for speech activity detection is derived from the acoustic model used for ASR. The acoustic model is generated from partitioning the acoustic space by context-dependent phonemes with the context defined in this work as plus and minus five phonemes, cross-word to the left only. The context-dependent phoneme observation generation process is modeled as a GMM within the hidden Markov model (HMM) framework, and in typical large-vocabulary ASR systems, this can easily lead to more than 1k states and 40k Gaussian mixture components. Calculating all HMM state likelihoods from all Gaussians at each frame would preclude real-time operation. Therefore, we define a hierarchical structure for the Gaussians, where it is assumed that only a small subset of them is significant to likelihood computation at any given time [12]. The hierarchical structure takes advantage of the sparseness by surveying the Gaussian pool in multiple resolutions given some acoustic feature vector \mathbf{x} . As part of the training process, the complete set of available Gaussian densities

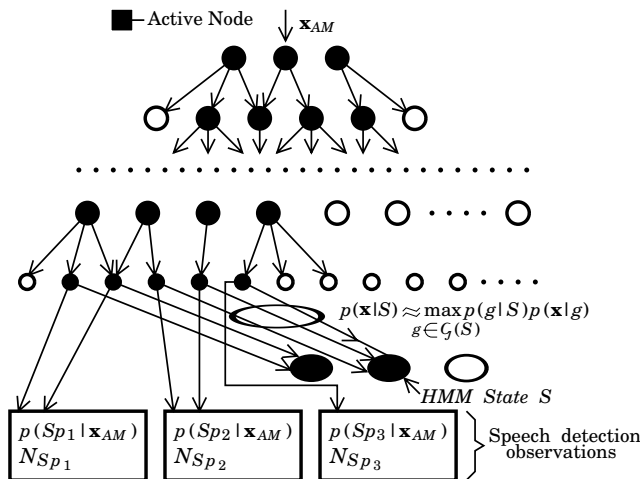


Fig. 2. Hierarchical acoustic model and the corresponding acoustic phonetic speech detection observations

is clustered into a search tree, in which the leaves correspond to the individual Gaussians, and a parent node is the centroid of its children for a defined distance metric. At the bottom of this tree resides a many-to-one mapping, collapsing the individual Gaussians to the appropriate HMM state. Therefore, the HMM state s conditional likelihood of a given observation vector \mathbf{x} at time t is computed as

$$p(\mathbf{x}|s) = \sum_{g \in \mathcal{G}(s)} p(g|s) p(\mathbf{x}|g),$$

where $\mathcal{G}(s)$ is the set of Gaussians that make up the GMM for state s . Traversing the tree yields a subset of active Gaussians, denoted by \mathcal{Y} . Based on \mathcal{Y} and the many-to-one mapping, the conditional likelihood of a state is approximated as

$$p(\mathbf{x}|s) = \max_{g \in \mathcal{Y} \cap \mathcal{G}(s)} p(g|s) p(\mathbf{x}|g).$$

If no Gaussian from a state is present in \mathcal{Y} , a default floor likelihood is assigned to that state.

To define the acoustic phonetic space used for speech activity detection, we apply an additional many-to-one mapping to the pruned result of the hierarchical tree. The mapping is based on grouping phonemes into three broadly defined classes: (i) the pure silence phoneme, trained from non-speech; (ii) the disfluent phonemes, which are noise-like phonemes, namely the unvoiced fricatives and plosives, i.e., the ARPAbet subset $\{/b/, /d/, /g/, /k/, /p/, /t/, /f/, /s/, /sh/\}$; and (iii) all the remaining phonemes, such as the vowels and voiced fricatives. The three classes will be denoted by Sp_1 , Sp_2 , and Sp_3 , respectively. Then, from the acoustic feature \mathbf{x} , used to traverse the acoustic model hierarchy, we can

form the speech detection class posteriors for the three speech detection classes as

$$Pr(Sp_i|\mathbf{x}) = \frac{1}{acc_mass} \sum_{g \in \mathcal{Y} \cap \mathcal{G}(Sp_i)} p(\mathbf{x}|g) p(g|Sp_i), \quad (7)$$

where

$$acc_mass = \sum_{i=1}^3 \left\{ \sum_{g \in \mathcal{Y} \cap \mathcal{G}(Sp_i)} p(\mathbf{x}|g) p(g|Sp_i) \right\},$$

and $\mathcal{G}(Sp_i)$ is the set of Gaussians defined by the mapping from phoneme to speech detection class Sp_i .

The process is illustrated in Fig. 2. Notice that the pruning at each level is accomplished using a threshold relative to the maximum scoring likelihood for that level [12]. As a result, the sharper the drop-off in Gaussian likelihoods, the more aggressive the pruning becomes. Therefore, both SNR and the phoneme being pronounced impacts the pruning. Features extracted from vowels and other voiced phonemes will result in more aggressive pruning than unvoiced fricatives, plosives and silence phonemes. This pruning will remain relative to SNR, with increasing SNR resulting in an overall more aggressive pruning.

The above observation results in additional speech detection features, based on class-normalized Gaussian counts. Denoting by N_{Sp_i} the number of Gaussians after hierarchical pruning that map to speech detection class Sp_i (see also Fig. 2), we consider the normalized counts

$$\bar{N}_{Sp_i} = N_{Sp_i} / \sum_{j=1}^3 N_{Sp_j}, \quad \text{for } i = 1, 2, 3, \quad (8)$$

as additional features. Combining (7) and (8) we obtain the six-dimensional acoustic phonetic feature space at frame t given by $v_a(t)$, as defined in (9):

$$\begin{aligned} v_{ai}(t) &= [\log(Pr(Sp_i|\mathbf{x})) \quad \log(\bar{N}_{Sp_i})] \\ v_a(t) &= [v_{a1}(t) \quad v_{a2}(t) \quad v_{a3}(t)] . \end{aligned} \quad (9)$$

3 SAD System Training

The SAD system training consists of two steps: the first step concerns training the acoustic model, whereas the second focuses on training the speech/silence classifier. Details are provided in the following subsections.

3.1 Acoustic Model Training

In order to generate the acoustic phonetic features described in Section 2.2, an acoustic model is required. Such a model is trained based on far-field lecture and meeting data, as described in an accompanying paper [9]. To summarize, this is a speaker-independent model based on 40-dimensional features generated

from an LDA projection, applied to a concatenation of nine consecutive 13-dimensional PLP acoustic observation vectors. Such observations are computed at 100 Hz from a Hamming windowed 25 ms speech segment, and are mean normalized on a per-speaker basis. The resulting acoustic model is composed of three-state, left-to-right HMM phonetic models, with the final model having a total of 6000 context-dependent states and approximately 200k Gaussians. The model is trained on 473.5 hrs of far-field data [9].

3.2 Speech/Silence Classifier Training

The fundamental classifier employed for speech/silence detection is a Gaussian mixture model (GMM). In this work, we investigate two modeling approaches, namely a diagonal GMM and a full covariance GMM. Details are provided next.

Feature Combination for the Diagonal GMM: From (6) and (9) we derive fused (concatenated) 11-dimensional features

$$v_f(t) = [v_e(t) \ v_a(t)] . \quad (10)$$

In order to de-correlate such features and allow classification by a diagonal-covariance GMM, we apply *principal component analysis* (PCA) to (10). In particular, we choose as subspace the basis set formed by the eigenvectors corresponding to the top eight eigenvalues. This results in the projected eight-dimensional feature vector

$$v_p(t) = \mathbf{A} v_f(t) , \quad (11)$$

where \mathbf{A} denotes the PCA matrix.

Feature Combination for the Full Covariance GMM: In order to accomplish full-covariance GMM training, features with high correlation need to be removed from the observation vector. In particular, in (6), we drop the mid-to-low energy track $m2l(t)$, yielding a four-dimensional energy based feature vector

$$v_{e,FC}(t) = [e(t) \ let(t) \ met(t) \ het(t)] .$$

Concerning the acoustic-phonetic features, it may not be immediately obvious from (9) that each observation pair in (7) and (8) are highly correlated. Keeping all features results in sharp models that don't generalize robustly. It was experimentally determined that class-normalized Gaussian counts (see (8)) outperform posteriors (7) as features for SAD. Therefore, the following seven-dimensional feature vector is used in conjunction with the full covariance GMM:

$$v_{FC}(t) = [v_{e,FC}(t) \ \log(\overline{N}_{Sp_1}) \ \log(\overline{N}_{Sp_2}) \ \log(\overline{N}_{Sp_3})] . \quad (12)$$

GMM Training: We subsequently train a three-class GMM classifier (for each speech detection class described in Section 2.2) on vectors (11) in the diagonal GMM case, or (12) in the full covariance case. GMM training is accomplished in two steps: First, we pool all training vectors – independently of the associated

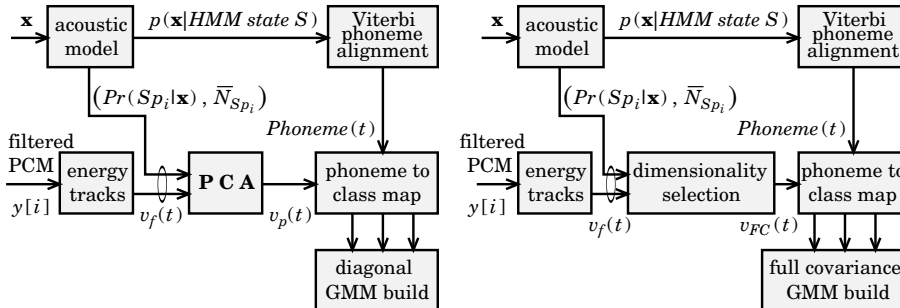


Fig. 3. Training the three-class speech detection classifier using diagonal (left) or full-covariance (right) GMMs

class labels – and run the LBG splitting algorithm [13], where the *expectation maximization* (EM) algorithm is iteratively run to convergence between splits. Splitting is terminated when the desired number of mixtures is reached: That number is eight in the case of the diagonal GMM, and two or four mixtures in the case of the full covariance models, as discussed in our experiments (Section 5). This initial step generates a class-independent GMM. Subsequently, the training vector class labels are obtained by Viterbi alignment, using the acoustic model and the phoneme-to-speech detection class mapping (see also Fig. 3). A single EM step is then run employing the individually pooled class-specific training vectors, with the class-independent GMM used as the starting model. The effectiveness of this process has been determined empirically and borrows from techniques found in speaker verification literature [14].

In particular, the GMMs were trained using the following resources: (i) All CHIL 2006 development data; (ii) All CHIL data within the RT05 development set; and (iii) Downsampled ICSI, NIST, RT04, non-CHIL RT05, and AMI data (see also [9]). Far-field microphone audio data were pooled together, resulting in approximately 19 hours of CHIL data and 4.7 hours from the additional sets.

4 SAD System Testing

Following the acoustic model and GMM training steps, we now proceed to describe how the SAD system is applied on single- and multi-channel audio signal inputs. Clearly, the first step is to extract the energy and acoustic features as described in Sections 2.1 and 2.2. Subsequently, the GMM is applied at each frame $t = n$ to features (11) or (12), for the diagonal or full covariance GMMs respectively. This results to the log-probability scores $L(Sp_i, n) = \log Pr(Sp_i|v_f(n))$, for each of the three classes discussed in Section 2.2. To obtain the final speech/silence intervals, these scores are further processed in the temporal domain by two stages of smoothing, the first of which also collapses the three possible classes into the two classes of interest: speech and silence. Finally, a third stage of processing is applied in the multiple distant microphone (MDM) evaluation condition. More details on these processing steps are given in the following.

4.1 Score Smoothing, Class Merging, and Score Integration

First, at each frame and for each of the three classes, we locally smooth scores $L(Sp_i, n)$ over a small fixed window of $w=10$ consecutive frames (i.e., a window of 100 ms duration). The resulting smoothed scores are

$$\bar{L}(Sp_i, n) = \frac{1}{w} \sum_{k=0}^{w-1} L(Sp_i, n - k).$$

We subsequently re-assign the “silence” class score at each frame, by merging the smoothed scores of the pure silence and disfluent classes by a simple max rule: $Score(silence, n) = \max_{i=1,2} \bar{L}(Sp_i, n)$. On the other hand, the “speech” score is initially set to the purely voiced class: $Score(speech, n) = \bar{L}(Sp_3, n)$. Following this step, the two scores are temporally accumulated as described next: When the global state is in “silence”, the condition $Score(speech, n) > Score(silence, n)$ results in accumulation of the scores to determine if the global state is to be switched to speech. Denoting by frame n^* the initial frame in the silence state such that $Score(speech, n^*) > Score(silence, n^*)$, we begin integrating the difference between the “speech” and “silence” scores, namely

$$\Delta_{Sp}(n^* + \delta) = \frac{1}{\delta + 1} \sum_{k=0}^{\delta} \left(\bar{L}(Sp_3, n^* + k) - \max_{i=1,2} \bar{L}(Sp_i, n^* + k) \right). \quad (13)$$

The global state is then changed to “speech”, once condition $\Delta_{Sp}(n^* + \delta) > 0$ is satisfied for any value $\delta \in [N_{min}, N_{max}]$. A similar procedure is used to switch from the “speech” to “silence” state, with the “speech” and “silence” scores swapped in (13). In our work, N_{min} and N_{max} are set to 50 and 100 ms respectively.

The above algorithm produces a segmentation where disfluent speech is lumped into the “silence” class. The last step is then to refine this segmentation in order to more accurately determine the true speech and silence boundaries. From the smoothed scores $\bar{L}(Sp_2, n)$, we know which regions were assigned to “silence” based on the disfluent class. Therefore, whenever a segment is classified as class Sp_2 , it is mapped to speech (Sp_3), only if it lies between segments Sp_1 (silence) to its left and Sp_3 (speech) to its right (the condition of changing the global state from silence to speech), or vice-versa (speech to silence transition region); otherwise it is mapped to silence (Sp_1). This is intended to handle consonant-vowel-consonant transitions within a word, while maintaining robustness to non-stationary noise.

4.2 Lead, Lag, and Silence-Collapsed Smoothing

The above procedure provides a first signal segmentation into speech and silence intervals. However, this tends to be “over-segmented”, and with very tight boundaries of the speech intervals. This necessitates a second level of smoothing that is driven by two temporal parameters, optimized based on development experiments. The first parameter, P_1 , is designed to expand the speech intervals by P_1 ms on either side. Following such padding, a second parameter, P_2 , is used to “collapse” silence segments that are of duration less than P_2 ms.

Table 1. “Speaker diarization error”, %, for single-channel (SDM) SAD on three development sets for various SAD GMMs and smoothing parameters P_1 and P_2

SAD system parameters				development sets		
GMM covariance	# mixtures	P_1 (ms)	P_2 (ms)	Dev_1	Dev_2	Dev_3
Diagonal	8	300	150	9.77	1.10	9.25
Diagonal	8	300	250	9.62	1.17	9.12
Full	2	300	150	9.94	2.27	9.49
Full	2	300	250	9.69	1.92	9.25
Full	4	300	150	9.02	1.46	9.02
Full	4	300	250	8.84	1.23	8.94

4.3 SAD System Combination in the MDM Condition

It is expected that, when multiple microphone signals are available, speech activity detection may become more robust. This of course requires an appropriate fusion approach to combine the available multi-channel information. In this work, we choose a “decision fusion” methodology that utilizes class-only information, namely the single-channel SAD system outputs. In particular, for each time frame, we consider a simple *majority rule*, applied on the set of all available microphone channel SAD outputs at the particular time frame. The rule is implemented to be biased towards speech in case of a tie, which of course can only occur if the number of available microphone channels is even. Notice that this approach assumes synchronicity among all channels, which in general holds for the CHIL data due to the data capture mechanism and the relatively small smart room size.

Based on the majority rule, we consider two variants of SAD system combination in conjunction with the second level of SAD output smoothing that was discussed earlier (Section 4.2). In the first approach, referred to as “*Rover A*” method, smoothing is first applied independently per channel (as in the single-channel SAD system), followed by the majority rule decision. The second variant employs the first level of smoothing as in single-channel SAD (Section 4.1), but interjects the majority rule multi-channel fusion before applying the second smoothing stage (Section 4.2) to the combined output. The latter will be referred to as the “*Rover B*” method. “*Rover A*” is the technique used in the IBM MDM SAD system submitted to the RT06s evaluation campaign.

5 Experimental Results

We now proceed to report SAD system experimental results. We first summarize system variant comparisons on development data, followed by evaluation results achieved on the RT06s campaign.

In the previous sections, we discussed a number of possibilities for SAD system training and testing. For example, diagonal or full-covariance GMMs, parameters P_1 , P_2 for SAD output smoothing, and two “*Rover*” variants for multi-channel

Table 2. SAD speaker diarization error, %, on development data, when using two channel combination techniques on the multiple distant microphone (MDM) condition. Results on a single-channel (SDM) are also depicted. In all cases, a four-mixture, full-covariance GMM with $P_1 = 300$ ms and $P_2 = 250$ ms is used.

Condition / Method	Dev_1	Dev_2	Dev_3
SDM	8.84	1.23	8.94
MDM – “Rover A”	8.61	0.56	8.57
MDM – “Rover B”	8.72	0.53	8.52

Table 3. RT06s evaluation results for the IBM SDM and MDM SAD systems. Speaker diarization error, % (as well as detailed FA/FR, %), using the initial ELDA and final CMU reference transcripts are depicted.

Condition	Total err. (FA/FR) (ELDA ref.)	Total err. (FA/FR) (CMU ref.)
SDM	12.15 (5.7/6.5)	8.62 (2.3/5.3)
MDM	8.02 (2.8/5.2)	5.01 (0.2/4.2)

combination. To choose the optimal approach, we conduct a number of experiments on development data. We utilize three sets for this purpose:

- Set Dev_1 : This set consists of seven seminars with a refined manual segmentation, kindly provided to us by CHIL partner UPC. The data include three seminars recorded at CHIL partner UKA, and one each at the IBM, ITC, AIT, and UPC sites. The segmentation is based on the original transcripts provided by CHIL partner ELDA, but only the UKA subset has been fully re-transcribed.
- Set Dev_2 : This is a subset of the previous set, consisting of the three fully re-transcribed UKA seminars. The speech/silence reference segmentation is therefore quite accurate.
- Set Dev_3 : This contains all CHIL dev. 2006 data, as segmented based on the ELDA transcripts.

It is important to note that the ELDA transcripts are well suited for ASR, but unfortunately are unreliable for SAD, as the speech/silence boundaries are not always accurately labeled. Thus, among the three development sets, only Dev_2 results can be fully trusted. However, the need for taking into account data from the other CHIL recording sites, when making system design decisions, forces us to also place emphasis on Dev_1 and Dev_3 , so as to avoid overfitting to UKA data characteristics.

A number of development set results are depicted in Tables 1 and 2. In Table 1, we depict development set results for three GMM models, one using eight diagonal-covariance mixtures employing eight-dimensional features, the other with two or four full-covariance mixtures using seven-dimensional features (see also Section 3.2). The three models require 136, 74, and 146 parameters to be es-

estimated, respectively. Based on this table, the four-mixture full-covariance model is chosen for the RT06s evaluation, since it achieves the best performance on two of the three sets, Dev_1 and Dev_3 . In addition, parameter values $P_1 = 300$ ms and $P_2 = 250$ ms are chosen, based on a number of experiments that are not reported here due to lack of space.

Next, we compare the two channel combination techniques discussed in Section 4.3 for MDM SAD. Results are depicted in Table 2 for the chosen SAD system parameters. Single-channel SAD results are also depicted for comparison purposes. Clearly, improvements are significant, especially for set Dev_2 that is accompanied by the most accurate reference segmentation. However, no significantly consistent difference can be observed between the two combination techniques. “Rover A” is finally the technique chosen for the RT06s evaluation.

Finally, the RT06s evaluation results achieved by the IBM SDM and MDM systems are depicted in Table 3 of the CHIL eval. 2006 data set. Results using two reference segmentations are depicted: The ones derived from the initial ELDA transcripts, as reported in the RT06s workshop in May 2006, and the finalized segmentation after re-transcription by a team from the Carnegie Mellon University (CMU) in early June 2006. The latter are the official evaluation results. Note the significant improvement in the MDM condition, compared to the SDM results.

6 Summary

In this paper, we presented a novel approach to speech activity detection that augments energy based features with acoustic phonetic ones. In contrast to traditional systems that often use the frequency based speech representation to directly provide features to a speech/silence classifier, the proposed technique utilizes an acoustic model to provide likelihood based features to the detector using a phoneme grouping into three clusters. The algorithm performed well in the RT06s evaluation campaign, where it was applied to speech activity detection based on both single- and multi-channel far-field input.

Acknowledgements

The authors would like to acknowledge support of this work by the European Commission under integrated project CHIL, “Computers in the Human Interaction Loop”, contract number 506909.

References

- [1] Macho, D., Padrell, J., Abad, A., et al., “Automatic speech activity detection, source localization, and speech recognition on the CHIL seminar corpus,” *Proc. ICME*, 2005.
- [2] Li, Q., Zheng, J., Zhou, Q., and Lee, C.-H., “A robust, real-time endpoint detector with energy normalization for ASR in adverse environments,” *Proc. ICASSP*, pp. 233–236, 2001.

- [3] Martin, A., Charlet, D., and Mauuary, L., "Robust speech/non-speech detection using LDA applied to MFCC," *Proc. ICASSP*, pp. 237–240, 2001.
- [4] Bou-Ghazale, S. and Assaleh, K., "A robust endpoint detection of speech for noisy environments with application to automatic speech recognition," *Proc. ICASSP*, pp. 3808–3811, 2002.
- [5] Padrell, J., Macho, D., and Nadeu, C., "Robust speech activity detection using LDA applied to FF parameters," *Proc. ICASSP*, vol. 1, pp. 557–560, 2005.
- [6] Monkowski, M., *Automatic Gain Control in a Speech Recognition System*, U.S. Patent US6314396.
- [7] Marcheret, E., Visweswariah, K., and Potamianos, G., "Speech activity detection fusing acoustic phonetic and energy features," *Proc. ICSLP*, 2005.
- [8] Chu, S.M., Marcheret, E., and Potamianos, G., "Automatic speech recognition and speech activity detection in the CHIL smart room," *Proc. MLMI*, pp. 332–343, 2005.
- [9] Huang, J., Westphal, M., Chen, S., et al., "The IBM rich transcription spring 2006 speech-to-text system for lecture meetings," *Proc. MLMI* (same volume), 2006.
- [10] Van Compernelle, D., "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," *Proc. ICASSP*, pp. 833–836, 1990.
- [11] Armani, L., Matassoni, M., Omologo, M., and Svaizer, P., "Use of a CSP-based voice activity detector for distant-talking ASR," *Proc. Eurospeech*, pp. 501–504, 2003.
- [12] Novak, M., Gopinath, R.A., and Sedivy, J., "Efficient hierarchical labeler algorithm for Gaussian likelihoods computation in resource constrained speech recognition systems," available on-line at: <http://www.research.ibm.com/people/r/rameshg/novak-icassp2002.ps>
- [13] Gersho, A., and Gray, R.M., *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 3rd Ed., Ch. 11, 1993.
- [14] Ramaswamy, G.N., Navratil, A., Chaudhari, U.V., and Zilca, R.D., "The IBM system for the NIST-2002 cellular speaker verification evaluation," *Proc. ICASSP*, vol. 2, pp. 61–64, 2003.