HAPPY Team Entry to NIST OpenSAD Challenge: A Fusion of Short-Term Unsupervised and Segment i-Vector Based Speech Activity Detectors

Tomi Kinnunen[1], *Alexey Sholokhov*[1], *Elie Khoury*[2], *Dennis Thomsen*[3], *Md Sahidullah*[1], *Zheng-Hua Tan*[3]

[1]University of Eastern Finland, Finland
[2]Pindrop, USA
[3]Aalborg University, Denmark
tkinnu@cs.uef.fi

Abstract

Speech activity detection (SAD), the task of locating speech segments from a given recording, remains challenging under acoustically degraded conditions. In 2015, National Institute of Standards and Technology (NIST) coordinated OpenSAD bench-mark. We summarize "HAPPY" team effort to OpenSAD. SADs come in both unsupervised and supervised flavors, the latter requiring a labeled training set. Our solution fuses six base SADs (2 supervised and 4 unsupervised). The individually best SAD, in terms of detection cost function (DCF), is supervised and uses adaptive segmentation with i-vectors to represent the segments. Fusion of the six base SADs yields a relative decrease of 9.3 % in DCF over this SAD. Further, relative decrease of 17.4 % is obtained by incorporating channel detection side information.

Index Terms: speech activity detection, NIST OpenSAD

1. Introduction

Speech activity detection (SAD) [1], the classic problem to locate speech segments from a given recording, finds use in coding [2] and recognition applications to prevent unnecessary processing of non-speech segments. A large number of SAD methods have been studied, ranging from rule-based digital signal processing methods to advanced machine learning techniques. Even if performing well on high-quality audio, state-of-the-art methods lack generalization power when faced with severely degraded, unforeseen acoustic conditions. An increased recent research effort has been devoted to SAD, specifically within the DARPA RATS program[1].

We summarize the effort of "HAPPY" team to the recent NIST OpenSAD challenge[2], consisting of the three co-authoring teams of this study. In contrast to commonly adopted deep neural nets or other methods requiring supervised training (see Section 2), we focus mostly on unsupervised SADs (Table 1). Some of the authors are faced with practical needs to integrate robust SADs to speaker verification platform intended to operate in different languages and a variety of logical and physical access control use cases[3] where we may lack large labeled development data. Thus, our preference is on SADs that work reasonably well across varied conditions but require little to no supervised training or parameter optimization.

Table 1: The individual SADs of the HAPPY team. Two are supervised (require labeled training data), the rest unsupervised. Systems 1 to 5 produce decisions every 10 ms, system 6 uses adaptive segmentation with i-vectors segment representation.

Site	Id	Method	Superv.	Frame	Ref.
UEF	S1	GMM (st. MFCC)	✓	✓	[3]
	S2	GMM (PLP)	×	✓	
	S3	GMM (PNCC)	×	✓	
	S4	Sohn *et al.*	×	✓	[6]
AAU	S5	rSAD	×	✓	[4]
Pindrop	S6	i-vectors	✓	×	[5]

Our work has two novel contributions. Firstly, we enhance and fuse our earlier unsupervised [3, 4] and supervised [5] SADs by alternative front-end features (Subsection 3.1). Secondly, we adopt a simple fusion scheme that uses a soft channel detector to weight the importance of each SAD on a recording-by-recording basis.

2. A Brief Review of Modern SADs

SADs can be divided into two broad categories, unsupervised and supervised ones, depending on whether they require labeled training data. **Unsupervised** SADs include standard real-time SADs such as the one used by G.729 codec [7]. These techniques combine a set of low-complexity, short-term features such as energy, zero-crossing rate [7], periodicity [8] or spectral divergence [9]. The feature values are then compared against fixed or adaptive thresholds to produce SAD segmentation. Another subclass of unsupervised SADs includes statistical model-based methods [6, 10, 11] that treat SAD as a hypothesis testing problem via parametric modeling of spectral coefficients.

The above unsupervised methods are often designed for real-time operation. But there are off-line applications, such as speaker diarization and forensic audio analysis, where delayed SAD decisions are acceptable. Hence, some unsupervised methods take benefit of information over long-duration buffers or even the full audio recording (that could be minutes or hours long). For instance, a method frequently used in text-independent speaker verification determines energy-based SAD

[1]http://www.darpa.mil/program/robust-atuomatic-transcription-of-speech

[2]NIST disclaimer: "NIST serves to coordinate the NIST OpenSAD evaluations in order to support speech activity detection research and to help advance the state-of-the-art in speech activity detection technologies. NIST OpenSAD evaluations are not viewed as a competition: as such, results reported by NIST are not to be construed, or represented, as endorsements of any participant's system, or as official findings on the part of NIST or the U.S. Government". Web page: http://www.nist.gov/itl/iad/mig/opensad_15.cfm

[3]https://www.octave-project.eu/

threshold via bi-Gaussian modeling [12] of log-energy or maximum energy over the whole utterance. A similar approach that combines multiple features is given in [13]. Other methods include use of utterance-specific speech and nonspeech codebooks trained using energy SAD labels [3], 4 Hz modulation energy [14], *a posteriori* signal-to-noise ratio (SNR) weighted energy distance [15] and unsupervised sequential GMM on Mel sub-bands [16].

The main benefit of unsupervised SADs is simplicity as no additional labeled datasets are required. Nevertheless, they can be dependent on appropriate balance of speech and nonspeech segments. These motivate the use of other major type, **supervised** SADs, that leverage from large supply of labeled off-line data to train SADs. These include Gaussian mixture models (GMMs) [17, 18, 19, 20], hidden Markov model (HMMs) with Viterbi segmentation [21], deep neural networks (DNNs) [22], recurrent neural network (RNNs) [23] and long short-term memory (LSTM) RNNs [24] to mention a few. A down-side, besides the requirement for additional data and high training complexity, is risk for over-fitting [25] unless the training utterances are representative enough of the actual operating conditions. This might be in part alleviated with the use of robust acoustic features, such as power-normalized cepstral coefficients (PNCCs) [20] or log-mel features [21, 26].

3. Individual SADs of the HAPPY team

3.1. Systems 1 to 3: GMM-based SADs

The first three subsystems use an approach similar to [3] (originally inspired by [27]), developed for speaker verification purposes [28]. It first trains speech and non-speech models using a small subset of MFCCs labeled automatically via an energy SAD, for a given utterance. Speech and non-speech codebooks are trained using k-means and all the frames are labeled using nearest-neighbor classification. The initial energy values are computed from a Wiener filter enhanced signal.

We revise the method in three ways. First, we use GMMs trained with maximum-likelihood instead of codebooks, leading to slightly increased accuracy (and increased execution time). We use 8 full-covariance Gaussians trained with 20 EM iterations. 10 % of the frames are used for initial labeling. Second, we use F0-based initialization instead of energy. We extract F0 using a cross-correlation method in the Snack Sound Toolkit [29]. When the number of detected frames was not sufficient, these segments were augmented by frames with the highest energy values taken from remaining part of an utterance.

Thirdly, we study the impact of front-end features. To this end, **System 1** uses stacked mel-frequency cepstral coefficients (MFCCs). We extract 13 MFCCs augmented with 15 preceding and following frames. The resulting (31×13)-dimensional supervector is then reduced to 50 dimensions using a modified linear discriminant analysis (LDA), trained using a random subset of 5% of files in the training data. In specific, we project data onto the eigenvectors of the matrix $\alpha \boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{S}_{\mathrm{B}} + (1-\alpha)\times\boldsymbol{S}_{\mathrm{T}}$, where $\boldsymbol{S}_{\mathrm{W}}$, $\boldsymbol{S}_{\mathrm{B}}$ and $\boldsymbol{S}_{\mathrm{T}}$ are the average within-class, between-class and the total scatter matrices. This is a heuristic combination of linear discriminant analysis (LDA) and principal component analysis (PCA) with the trade-off parameter α set to 0.95 throughout our experiments. System 1 is an *almost* unsupervised one – the only part requiring supervised training is the dimensionality reduction part but the SAD decision rule itself, trained separate for each file, is test-data driven.

System 2 uses 13-dimensional perceptual linear prediction (PLP) coefficients followed by RASTA filtering, extracted using [30], while **System 3** uses power-normalized cepstral coefficients (PNCCs) [31] that have shown promise in various recognition tasks. We use a publicly available implementation[4]. Systems 2 and 3 are completely unsupervised. Speech enhancement before extracting features was helpful for stacked MFCCs, but biased the PNCC and PLP systems towards high false alarm rates.

3.2. System 4: Statistical Model SAD

System 4 is the well-known statistical model SAD introduced in [6]. SAD decision is made via geometric mean of the likelihood ratios of individual frequency bands. The method uses a hangover scheme which considers the previous observations. We have used voicebox[5] implementation with *minimum statistics* (MS) noise tracker [32]. We set speech probability threshold to 0.25 considering the cost function of NIST OpenSAD (see Subsection 5.1).

3.3. System 5: rSAD

System 5, "rSAD" (robust speech activity detector) [4], [33], is also unsupervised. First, input signal is filtered by a first-order high-pass filter with a cutoff frequency of 60Hz, and *a posteriori* signal-to-noise-ratio (SNR) weighted energy difference is applied to detect high-energy segments. If the difference measure between two consecutive frames exceeds a predefined threshold, the frames are detected as high-energy ones. Consecutive high-energy frames are then grouped to form high-energy segments. Within a high-energy segment, if no pitch is detected, the corresponding segment is considered as noise. Secondly, a modified MS noise estimator [32] is used to remove the relatively stationary noise from the speech signal, and the high-energy noise segments are set to zero, generating a denoised signal. Concerning modifications of MS, we omit updating the noisy estimate during high-energy noise segments to avoid overestimation of the noise. In addition, the power of a frame is set to 0 if less than 5 frames in a 21-frame window (centered around the frame) have a power estimate that is less than half of the corresponding noise power estimate. In the end, the *a posteriori* SNR weighted energy difference measure is applied to the denoised signal, more specifically, extended speech segments containing pitch, to make speech activity detection. The source code for rSAD is publicly available[6].

3.4. System 6: Segment i-vector SAD

System 6 operates at longer cluster (segment) level as opposed to frame level. It consists of *generalized likelihood ratio - Bayesian information criterion* (GLR-BIC) based segmentation. The segments are represented using an i-vector [34], approach originally introduced for speaker verification and recently adopted for SAD in [5].

GLR-BIC segmentation: Inspired by prior work on speaker diarization [35, 36, 37], the aim is to split the audio recording into a set of homogeneous segments S_i and then merge the most similar segments in a hierarchical agglomerative manner. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_X}\}$ be a sliding window (e.g. $N_X = 100$) of feature vectors and $\mathcal{M}$ its parametric model. Assuming $\mathcal{M}$ to be a multivariate Gaussian, the generalized likelihood ratio (GLR) [35] at a possible point of change,

[4] http://www.cs.cmu.edu/~./mharvill/RATS/software_releases/PNCC/PNCC_deployed_v6/

[5] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[6] http://kom.aau.dk/~zt/online/rVAD/

c, is expressed (in log scale) by:

$$R(c) = \frac{N_X}{2}\log|\mathbf{\Sigma}_X| - \frac{N_{X_{1,c}}}{2}\log|\mathbf{\Sigma}_{X_{1,c}}| - \frac{N_{X_{2,c}}}{2}\log|\mathbf{\Sigma}_{X_{2,c}}| \quad (1)$$

where $\mathbf{\Sigma}_X$, $\mathbf{\Sigma}_{X_{1,c}}$ and $\mathbf{\Sigma}_{X_{2,c}}$ are the covariance matrices and N_X, $N_{X_{1,c}}$ and $N_{X_{1,2}}$ are the number of feature vectors in $\boldsymbol{X}$, $\boldsymbol{X}_{1,c}$ and $\boldsymbol{X}_{2,c}$, respectively. The GLR curve obtained by sliding the search window is further smoothed using the so-called Savitzky-Golay filter [38]. By maximizing the likelihood, the estimated point of change $\widehat{c}_{\text{glr}}$ is $\widehat{c}_{\text{glr}} = \arg\max_c R(c)$. These candidates of points of change are filtered out and adjusted using Bayesian information criterion (BIC) [36]. The new segments boundaries are estimated as follows:

$$\widehat{c}_{\text{bic}} = \arg\max_c \Delta\text{BIC}(c) \quad (2)$$

where $\Delta\text{BIC}(c) = R(c) - \lambda P$ and preserved if $\Delta\text{BIC}(\widehat{c}_{\text{bic}}) \geq 0$. Finally, the resulting segments are grouped by hierarchical agglomerative clustering using the same BIC distance measure [39]. For more details, readers are invited to read [5].

I-vector based SAD: Any resulting cluster C of the above GLR-BIC segmentation contains mostly only speech or non-speech frames. Total variability modeling aims to extract low-dimensional i-vectors $\boldsymbol{\omega}$ from samples in C, using the well-known i-vector extractor [34] expression $\boldsymbol{\mu} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{\omega}$, where $\boldsymbol{\mu}$ is the supervector of C, $\boldsymbol{m}$ is the supervector of universal background model and $\boldsymbol{T}$ is the low-rank total variability subspace matrix. Once the i-vectors are extracted, whitening and length-normalization [40] are applied for channel compensation purposes. For scoring, we use SVM [41] with a radial basis function (RBF) kernel. Platt scaling [42] is used to transform SVM scores into probability estimates. For SVM training, the ground-truth segmentation contains only speech or non-speech, while a segment might contain both speech and non-speech.

4. System Fusion

4.1. Score Fusion

Fusion of different SAD outputs is done using *logistic regression*, approach that is frequently employed for combining speaker classifiers [43, 44]. The fusion weights are trained using frame-wise cross-entropy. Let a frame O_t be processed by N_s SAD systems. Each system produces an output score denoted by $h_s(O_t)$. Two fusion strategies were explored. The first one uses only the output scores. It is expressed by the logistic function:

$$f_1 = g\left(\alpha_0 + \sum_{s=1}^{N_s}\alpha_s h_s(O_t)\right) \quad (3)$$

where $g(x) = 1/\left(1+\exp(-x)\right)$ and $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_N]$ denote the fusion weights. The second strategy (inspired by [45]) integrates channel side information, with the idea that certain SADs might perform better than others on certain types of channels. To this end, we extend Eq. 3 by:

$$f_2 = g\left(\alpha_0 + \sum_{s=1}^{N_s}\alpha_s h_s(O_t) + \sum_{s=1}^{N_s}\sum_{c=1}^{N_c}\beta_{s,c} h_s(O_t) p_c(O_t)\right) \quad (4)$$

where $p_c(O_t)$ is the posterior probability of O_t being generated by channel c and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ the regression parameters. We optimize the fusion parameters using a conjugate gradient optimizer [46]. In the following subsection we detail computation of the channel posteriors $p_c(U_t)$.

System 6 produces variable-duration segments obtained through segmentation while the remaining SADs use fixed frame rate. To enable fusion of the segment i-vector system with the frame-based systems, its SAD score given for every segment was copied to all the frames within the segment.

Table 2: Channel information as provided by NIST.

Channel	Frequency Band	Modulation Type
A	UHF	Narrow-band (NB) FM
B	UHF	NB FM
C	UHF	NB FM
D	HF	Single side-band (SSB) AM
E	VHF	NB FM
F	UHF	Frequency-hopping spread-spectrum
G	UHF	Wide-band (WB) FM
H	HF	AM
XA	HF	SSB AM
XH	UHF	WB FM
XI	HF	SSB with Digital Noise Reduction
XK	unknown	NB FM with co-channel interference
XN	unknown	NB FM

4.2. Channel Detection for Fusion Side Information

Our channel detector uses GMMs to produce channel posterior estimates. We adopt 13 MFCCs without any feature normalization or deltas, to make them maximally sensitive to channel, extracted using [30]. Using all the data in the training set, we first train 21 individual GMMs, each with 512 diagonal covariance Gaussians using all the data from each language-channel combination (i.e. $\{\texttt{eng}, \texttt{alv}, \texttt{urd}\} \times \{\texttt{B}, \texttt{D}, \texttt{E}, \texttt{F}, \texttt{G}, \texttt{H}, \texttt{src}\}$) (See Subsection 5.1 for details). For each frame, we first compute the individual model posterior probabilities and then sum the probabilities up over the languages to get 7 channel posteriors per frame (another option, not studied here, would have been to simply pool all the training data to train 7 channel GMMs). Results on dev part yield frame-level channel detection error rate of 11.22% (over about 240 million frames), over the known channels, and file-level error rate of 1.008%. To obtain this latter result, the frame-level channel posteriors were averaged for a given file.

We assume a closed (known) set of training channels. For the known channels, we found the posterior vectors to peak sharply at the correct channel. In this case (4) "selects" a column from the matrix $\boldsymbol{\beta}$. In the case of an unforeseen or uncertain channel, we assume it can be represented as a combination of the known channels.

5. Experiments and Results

5.1. Data and Evaluation Metric

The NIST 2015 OpenSAD evaluation uses data collected for the DARPA RATS program. The organizers divided their data into three parts, **Training**, **Development** and **Evaluation**, all consisting of re-transmitted telephone conversations captured through different communication channels (see Table 2). The source audio originated from existing LDC corpora (Fisher English, Fisher Levantine Arabic and CALLFRIEND Farsi) and data collected in the RATS program. Training and Development parts include speech in five languages: Levantine Ara-

bic (alv), American English (eng), Farsi (fas), Pashto (pus) and Urdu (urd). Channels A and C were excluded from the Training and Development parts by the organizers. The Development set provided by NIST was divided into two parts, dev-1 and dev-2, the latter having additional channels. Since the ground-truth annotations were deemed reliable by NIST only for channels B, D, E, F, G and H in dev-2, we excluded channels XA through XN from dev-2 as well as whole dev-1, resulting in a reduced development set of 661 files. Source files (src) were also excluded. This reduced dev-set (see Table 3) was used for optimizing the base SADs and the fusion parameters.

Table 3: Data used by HAPPY team for system development.

Part	Language					Channels	Total files
	alv	eng	urd	fas	pus		
Train	✓	✓	✓			{B-H, src}	5485
Dev	✓		✓	✓	✓	{B-H}	661

The official evaluation metric of the NIST OpenSAD challenge is the detection cost function, DCF $= \gamma P_{\text{fa}} + (1-\gamma) P_{\text{miss}}$, where P_{fa} is the false alarm rate (proportion of non-speech frames mis-classified as speech) and FRR is the miss rate (proportion of speech frames mis-classified as non-speech). The weight $\gamma = 0.25$ penalizes missed speech detections more heavily. The DCF values are computed per file and averaged. A global channel-independent threshold was applied.

5.2. Results

Table 4 shows the results on the devset. In terms of DCF, systems 5 and 6 outperform the GMM-based and Sohn methods. Comparing the GMM-based systems, system 2 (PLP features) and 3 (PNCCs) outperform system 1 that uses stacked MFCCs.

Fusion generally helps, as expected. Both fusion strategies decrease DCF though neither P_{fa} nor P_{miss} separately achieve the corresponding column minima. The first fusion decreases DCF by 9.3 % relative over the best individual system (sys 6). The second fusion strategy, utilizing the channel side information, yields a relative reduction of 17.4 % in DCF over sys-6, suggesting that channel side information is useful. Table 5 shows the fusion weights from fusion rule (3). As expected, the relative weight magnitudes agree with the respective DCFs, though for instance system 5 has the largest contribution while system 6 is individually best. Even the low-performing systems 1, 2 and 4 have non-zero weights.

Table 6 further compares the best individual (sys 6) and fusion of all SADs (last row) to two simplified fusions: (1) fusion of the two best SADs (systems 5 and 6) in the second row, and (2) fusion of mostly-unsupervised SADs (systems 1 to 5) in the third row. Fusing systems 5 and 6 provides a relative reduction of about 12 % over system 6. Fusing systems 1 to 5 gives a DCF very close to system 6, suggesting that many unsupervised SADs fused may reach accuracy close to one supervised one. Comparing rows 2 and 4, DCF *increases* 3 % relative when all the six SADs are fused. Thus, the simpler combination (systems 5 and 6) is more attractive. In principle, fusion of more systems should not hurt but as the fusion training objective is not the same as the evaluation metric, this can happen.

The fusion system that formed the primary submission of the HAPPY team corresponds to Fusion-2 (Eq. (4)) that combines all six SADs with the channel side information. The official results released by NIST on the eval-set are given in Table 7 for different forgiveness collar sizes. As expected by definition, DCF and P_{fa} decrease by increased collar size, while P_{miss} remains unchanged since the collar affects non-speech parts only. At this stage, since NIST has not released the eval-set key, we are unable to conduct further analyses on eval-set.

Table 4: Results on Dev data with a collar size of 2 sec.

System	P_{miss}	P_{fa}	DCF
Sys-1	0.0615	0.4317	0.1540
Sys-2	0.0654	0.2257	0.1054
Sys-3	0.0772	0.1714	0.1008
Sys-4	0.0635	0.3889	0.1449
Sys-5	0.0478	0.0575	0.0502
Sys-6	0.0277	0.1009	0.0460
Fusion-1 (all SADs) [Eq. 3]	0.0317	0.0720	0.0417
Fusion-2 (all SADs) [Eq. 4]	0.0294	0.0638	0.0380

Table 5: Fusion weights in Eq. [(3)]

S1	S2	S3	S4	S5	S6
0.1081	-0.1122	0.31681	-0.1162	1.85039	1.5674

Table 6: Similar to Table 4 but for reduced sets of fused SADs.

System	P_{miss}	P_{fa}	DCF
Sys-6 only	0.0277	0.1009	0.0460
Fusion-1 (S5 & S6) [Eq. 3]	0.0312	0.0691	0.0406
Fusion-1 (S1 to S5) [Eq. 3]	0.0309	0.0949	0.0469
Fusion-1 (all SADs) [Eq. 3]	0.0317	0.0720	0.0417

6. Conclusion

HAPPY team entry to NIST OpenSAD challenge consisted mostly of unsupervised SADs. Segment i-vector and rSAD worked the best and fusing all six SADs yielded further gain. Soft channel detection was useful on the dev-set (though failed to generalize; more robust fusion deserves further attention).

7. Acknowledgements

This paper reflects some results from the OCTAVE Project (#647850), funded by the Research European Agency (REA) of the European Commission, in its framework programme Horizon 2020. The views expressed in this paper are those of the authors and do not engage any official position of the European Commission. Part of the study was also funded by Academy of Finland (projects 283256 and 288558).

8. References

[1] J. Ramirez *et al., Voice Activity Detection. Fundamentals and Speech Recognition System Robustness*. InTech, June 2007.

[2] A. Kindoz and A. M. Kondoz, *Digital Speech; Coding for Low Bit Rate Communication Systems*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

Table 7: Official results on the evaluation data

Collar	P_{miss}	P_{fa}	DCF (official)
2 sec	0.0152	0.0782	0.0310
1 sec	0.0152	0.0931	0.0347
0.5 sec	0.0152	0.1244	0.0425
0.25 sec	0.0152	0.1784	0.0560
no collar	0.0152	0.2783	0.0810

[3] T. Kinnunen and P. Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data," in *Proc. of ICASSP*, Vancouver, Canada, May 2013, pp. 7229–7233.

[4] Z.-H. Tan and B. Lindberg, "Low-complexity variable frame rate analysis for speech recognition and voice activity detection," *IEEE J. Sel. Topics in Signal Proc.*, vol. 4, no. 5, pp. 798–807, Oct 2010.

[5] E. Khoury and M. Garland, "I-vectors for speech activity detection," in *Proc. Odyssey*, Bilbao, Spain, 2016, pp. 334–339.

[6] J. Sohn *et al.*, "A statistical model-based voice activity detection," *IEEE Sign. Proc. Lett.*, vol. 6, no. 1, pp. 1–3, Jan 1999.

[7] A. Benyassine *et al.*, "ITU-T recommendation G729 Annex B: A silence compression scheme for use with g729 optimized for v.70 digital simultaneous voice and data applications," *IEEE Communic. Mag.*, vol. 35, pp. 64–73, 1997.

[8] R. Tucker, "Voice activity detection using a periodicity measure," *Communications, Speech and Vision, IEE Proceedings I*, vol. 139, no. 4, pp. 377–380, 1992.

[9] J. Ramirez, J. Segura, C. Benitez, A. D. L. Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Comm.*, vol. 42, pp. 3–4, 2004.

[10] J. Shin, J.-H. Chang, and N. Kim, "Voice activity detection based on statistical models and machine learning approaches," *Comput. Speech Lang.*, vol. 24, no. 3, pp. 515–530, Jul. 2010.

[11] ——, "Voice activity detection based on a family of parametric distributions," *Pattern Recogn. Lett.*, vol. 28, no. 11, pp. 1295–1299, Aug. 2007.

[12] F. Bimbot *et al.*, "A tutorial on text-independent speaker verification," *EURASIP J. Appl. Sig. Proc.*, vol. 2004, no. 4, pp. 430–451, 2004.

[13] S. O. Sadjadi and J. H. L. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 197–200, 2013.

[14] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *IEEE ICASSP*, vol. 2, 1997, pp. 1331–1334.

[15] Z.-H. Tan and B. Lindberg, "Low-complexity variable frame rate analysis for speech recognition and voice activity detection," *IEEE J. Sel. Topics in Sig. Proc.*, vol. 4, no. 5, pp. 798–807, 2010.

[16] D. Ying, Y. Yan, J. Dang, and F. Soong, "Voice activity detection based on an unsupervised learning framework," *IEEE T. Audio, Speech, and Lang. Proc.*, vol. 19, no. 8, pp. 2624–2633, 2011.

[17] M. J. Alam *et al.*, "Supervised/unsupervised voice activity detectors for text-dependent speaker recognition on the RSR2015 corpus," in *Odyssey*, 2014, pp. 123–130.

[18] T. Hain and P. C. Woodland, "Segmentation and classification of broadcast news audio," Sydney, Australia, November 1998.

[19] L. Ferrer, M. McLaren, N. Scheffer, Y. Lei, M. Graciarena, and V. Mitra, "A noise-robust system for NIST 2012 speaker recognition evaluation," in *Proc. INTERSPEECH*, 2013.

[20] M. McLaren, M. Graciarena, and Y. Lei, "Softsad: Integrated frame-based speech confidence for speaker recognition," in *Proc. ICASSP*, 2015, pp. 4694–4698.

[21] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, and B. Kingsbury, "The IBM speech activity detection system for the DARPA RATS program." in *INTERSPEECH*, 2013, pp. 3497–3501.

[22] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks," in *Proc. Interspeech*, 2013, pp. 728–731.

[23] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *IEEE ICASSP*, 2013, pp. 7378–7382.

[24] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with LSTM recurrent neural networks and an application to hollywood movies," in *Proc. ICASSP*, 2013, pp. 483–487.

[25] X. Zhang and J. Wu, "Transfer learning for voice activity detection: A denoising deep neural network perspective," *CoRR*, vol. abs/1303.2104, 2013.

[26] S. Thomas, G. Saon, M. Van Segbroeck, and S. Narayanan, "Improvements to the IBM speech activity detection system for the DARPA RATS program," in *Proc. ICASSP*, 2015.

[27] M. Huijbregts *et al.*, "Filtering the unknown: Speech activity detection in heterogeneous video collections," in *Proc. of Interspeech*, Antwerp, Belgium, 2007, pp. 2925–2928.

[28] R. Saeidi *et al.*, "I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification," in *INTERSPEECH*, Lyon, August 2013, pp. 1986–1990.

[29] The Snack sound toolkit. http://www.speech.kth.se/snack/.

[30] D. P. W. Ellis, "PLP and RASTA (and MFCC, and inversion) in Matlab," 2005, online web resource. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

[31] C. Kim and R. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*, March 2012, pp. 4101–4104.

[32] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *Speech and Audio Processing, IEEE T.*, vol. 9, no. 5, pp. 504–512, Jul 2001.

[33] O. Plchot *et al.*, "Developing a speaker identification system for the DARPA RATS project," in *Proc. ICASSP*, May 2013, pp. 6768–6772.

[34] N. Dehak *et al.*, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[35] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *Proc. ICASSP*, vol. 2, Toronto, Canada, May 1991, pp. 873–877.

[36] S. Chen and P. Gopalakrishnan, "Clustering via the Bayesian information criterion with applications in speech recognition," in *Proc. ICASSP*, vol. 2, 1998, pp. 645–648.

[37] E. Khoury, C. Senac, and J. Pinquier, "Improved speaker diarization system for meetings," *Proc. ICASSP*, pp. 4097–4100, 2009.

[38] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[39] C. Barras *et al.*, "Improving speaker diarization," in *Proc. of DARPA RT04*, Palisades, USA, 2004.

[40] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[41] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.

[42] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. MIT Press, 1999, pp. 61–74.

[43] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the NIST'99 1-speaker submissions," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 237–248, 2000.

[44] N. Brümmer *et al.*, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE T. Audio, Speech & Lang. Proc.*, vol. 15, no. 7, pp. 2072–2084, September 2007.

[45] M. Mandasari *et al.*, "Quality measure functions for calibration of speaker recognition systems in various duration conditions," *IEEE T. Audio, Speech, and Lang. Proc.*, vol. 21, no. 11, pp. 2425–2438, 2013.

[46] T. P. Minka, "Algorithms for maximum-likelihood logistic regression," CMU Statistics Department, Tech. Rep. 758, 2001.