Voice Activity Detection for Children's Read Speech Recognition in Noisy Conditions

Ankita Pasad, Kamini Sabu, Preeti Rao
Department of Electrical Engineering,
Indian Institute of Technology Bombay, India
{ankitapasad,kaminisabu,prao}@ee.iitb.ac.in

***Abstract*—Recordings of read-aloud stories by children in a school setting can be used to provide an assessment of reading skills via automatic speech recognition (ASR). ASR, however, is known to be highly susceptible to background noise. The unusual variety of foreground (breath release, mic pops, etc.) and background (children playing, distinct background talker, wind, etc.) non-speech sounds makes this application particularly challenging. Motivated by the observation on real-world data that close to 50% of the recorded audio comprises purely non-speech activity, we investigate robust approaches to voice activity detection to eliminate non-speech segments to the extent possible prior to ASR. We have exploited energy-based and harmonicity-based features coupled with suitable temporal smoothing constraints in a two-pass noise preprocessing system. A discussion of the voice activity detection performance of the system is presented with reference to the characteristics of the noise types.**

I. Introduction

It is well known that in India's large rural population, millions of children complete primary school every year without achieving even basic reading standards[1]. Since reading competence enhances overall learning by enabling the child to self-learn various subject material from the vast available text resources, the importance of imparting reading skills in early school cannot be overstated. Automatic literacy assessment is one of the promising applications of ASR systems. It will make learning more interactive and teaching more efficient with reliable feedback from the assessment system. A story reading task based Reading Tutor App [2] on an Andriod tablet, has been deployed in one such resource-constrained rural region near Dahanu Taluka in Maharashtra for 10 to 14-year-old students, which also serves the purpose of task-specific data collection for the LETS project [3][4].

Over the past few years, there has been a considerable progress in the field of ASR. This has resulted in highly accurate performance for specific tasks in constrained environments. However, noise has been long since known to be deleterious to ASR performance unless rightful measures are employed. Unaccounted non-speech segments used in training can corrupt the acoustic models, and can further lead to increase in misdetections during recognition. In a recording environment, like ours, unaccounted conditions, spontaneous speaking styles, and the quality of the recording microphone can also distort the speech signal. The data obtained from rural settings is susceptible to a variety of noise due to an uncontrolled acquisition process. Noisy data adds to the train-test mismatch, and thus it is really important to appropriately deal with it either before or during the recognition task. A robust ASR system is an absolute necessity for our task as a single false negative can demotivate the child and hamper his/her trust in the tutor application.

There exist a variety of methods in the literature[5] which deal with noise at the ASR system level: (i) robust feature extraction, (ii) model adaptation, (iii) multi-condition training, (iv) joint model training. The presence of many varieties of noise with limited data makes noise/noisy speech modeling techniques critical to the task. Long non-speech regions deteriorate the ASR performance with too many word insertions as we don't have good noise models. This motivates the need to focus on non-speech segments detection so that most of the irrelevant information can be thrown out in the pre-processing part and ASR system can focus more on phonetic acoustic models without worrying about modeling every kind of noise.

In this work, we have focused on tackling noisy/non-speech data at the pre-processing level. We also come across a lot of unusable recordings due to extreme levels of noise in the background and it would be helpful to implement an online method which detects noisy environment and flags it. This will ensure that reader's effort is not wasted.

The organization of the paper is as follows. Section II explains the database under consideration in this study and annotation guidelines used for evaluation purpose. Section III categorizes and provides a detailed review of existing methods for noise pre-processing. Section IV proposes a way of handling selected types of non-speech sounds. Section V discusses the results of the proposed method. Section VI presents conclusion and future work.

II. Noise Characteristics and Dataset

The database collected for the project exhibits the presence of a variety of foreground and background non-speech sounds. These recordings are collected at 16 kHz sampling rate using a headset mic connected to the tablet. All the proceeding discussion and analysis is done on 19 audio files with an average duration of 142 seconds, which are believed to be good representatives of all types of noise and have been labeled manually for the ground truth decision. The major ones have been listed in table I. School noise is a broad category which includes distant speakers, children playing,

978-1-5090-5356-8/17/$31.00 © 2017 IEEE

paper shuffling, distant indistinct sounds. Background talker is a term used for background speaker who is distinctly audible, and babble refers to one or more distant speaker(s). Rain, school noise, generator, children playing sounds are always relatively long-lasting, whereas, bell, mic noise, breath release are always intermittent. Rest of the types may fall into either category. Table II presents a detailed distribution in terms of duration of non-speech sounds in 4 out of 19 files and the diversity of noise-types across different recordings is visible. Breath release and wind have been clubbed as they have similar characteristics. Bell and mic type sounds have not been accounted for in this table, they constitute 2-4% on an average. But, given the intermittent and time-localised nature of such sounds, low representation does not imply a low number of instances of such sounds. In the presence of overlapping noise types, only the most significant type was accounted for, based on the pre-defined set of priority rules. The following observations have been made from the noise characteristics distribution across the dataset:

- 51% of the test data is comprised of non-speech segments
- Most common noise types are rain, school noise, mic pops, breath release

Noise Type	**Characteristics**
BT	Speech-like
Bell	High pitch, and stationary
Rain	White noise-like spectral characteristics
Mic pop	Dense spectrum for a very short time duration
Babble	Speech-like, less prominent than BT
Generator	Low pitch, steady harmonics with constant amplitude
Wind, BR	Varying amplitude, no pitch
School noise	Highly varying amplitude as well as spectral characteristics

TABLE I: Characteristics of various noise types encountered in the database. BT: Background Talker, BR: Breath Release.

Audio Duration (s)	Rain (%)	SN (%)	BR,W (%)	BT (%)	Speech (%)
130.8	7	28	2	0	57
149.4	54	2	0	0	43
158	0	20	22	0	39
135.3	0	38	0	27	35

TABLE II: Categorization of representative test database as per non-speech sound types.SN: School Noise, BR: Breath Release, W: Wind, BT: Background Talker.

III. Literature Review

Voice Activity Detection (VAD) refers to the problem of distinguishing foreground speech segments from background noise in an audio stream. Along with segregating speech and non-speech regions, it is also important to appropriately deal with speech segments which have overlapping background noise. The more we can deal with various noise types at preprocessing step, the better it is for ASR performance. But, it has also been observed that speech enhancement techniques introduce distortions[6], which degrade recognition accuracy. So, our focus in the present work is to segregate the non-overlapping speech and noise regions before we delve into speech enhancement techniques.

Signal parameters like energy[7], ZCR[7], cepstral coefficients[8], wavelet[9] and spectral features are used for differentiating speech and non-speech. These parameters are usually checked against some fixed or adaptive threshold[7], the latter being more robust against changes in environmental noise levels. Its evaluation is specific to the cocktail party problem. Another measure, long-term spectral variability (LSTV), computed over long windows, has been recently used which compares stationarity of the speech and non-speech signal[10][11]. Different long-term properties of speech (spectral shape, spectro-temporal modulations, harmonicity and long-term spectral variability) have been compared using MLP classifier[12]. Some other machine learning methods like artificial neural network, sequential GMM, deep belief neural network, boosted DNNs are also used for VAD. These methods are known to improve performance at the cost of the quantity of training data required, and dealing with unexpected noise types becomes challenging as they need to be trained on a good amount of noise-specific data.

Usually, strong thresholding of energy tends to reject weak speech clippings. Different VAD methods nowadays are accompanied by hangover schemes to help retain the low energy starting and trailing segments. Conventionally, hang-over algorithms usually adopt a scheme that delays the transition from speech state to noise state[13]. The idea that consecutive occurrences of speech frames exhibit strong correlation, has been expressed by modeling the sequence of frame states as first-order Markov process in [14].

Robust VADs are also used in standard algorithms for speech compression and coding, like G.729 [15], AMR1 [16] and AMR2 [16]. G.729 uses four different parameters - the full and low-band frame energies, the set of line spectral frequencies (which are Linear Predictive Coding (LPC) coefficients), and zero crossing rates compared against long-term running averages. AMR (Adaptive Multi Rate Audio Codec) compares long-term energy estimates at different frequency sub-bands and also checks the presence of pitch or tone in these subbands. AMR being one of the state-of-the-art methods, we have used AMR2-VAD results as the benchmark for evaluation of proposed algorithm.

Given a limited dataset and a large variety of noise types, we have preferred the heuristic-based methods. As our data is collected using a close-talk microphone, the speech segments tend to have comparatively high energy. So, we have implemented an energy-based algorithm with adaptive threshold[17]. Along with this, we have incorporated a harmonicity-based method[18]. We were eventually benefitted by combining results of these two complementary parameters through a two-pass system.

IV. Implementation of VAD

A typical VAD system consists of two core parts: a feature extraction and a speech/non-speech decision mechanism. The first part extracts a set of parameters from the signal, which are used by the second part to make the final decision, based on a set of rules. Most of the VAD algorithms discussed in section III have targetted a specific noise type/characteristics, but given the diverse nature of non-speech sounds in our task, we are compelled to use a combination of features in order to come up with a generic algorithm. Here on, we refer to speech(1) detected as non-speech(0) as misdetections and non-speech(0) detected as speech(1) as false alarms. We have aimed to attain a low false alarm rate at a misdetection rate tolerance, decided by our benchmark algorithm. All the decisions are made on 10 milliseconds frame length.

A. Adaptive Linear Energy-based Detector (ALED)

Sakhnov et al.[17] have proposed an energy-based method which adapts the energy threshold on the occurrence of every detected non-speech frame. The threshold is updated as a weighted average of current threshold and most recently classified non-speech frame energy. The weight depends on the change in noise variance across time as observed from the ratio $r = \frac{\sigma_{new}}{\sigma_{old}}$. Here, σ_{new} and σ_{old} are noise variance values calculated from a fixed length buffer of recently detected non-speech frames. More the ratio r, more the change in noise variance and hence more weight is given to the energy of the current frame. The exact value of weight, p, used is as per the look-up table III. The threshold thus adapts according to the varying noise statistics. The threshold is initialized as an average of the frame-level short time energy for the first 4 seconds of the audio. For our task, it is reasonable to assume that first 4 seconds will always be background noise.

Ratio	**Value of p**
$r \geq 1.25$	0.25
$1.25 > r \geq 1.10$	0.20
$1.10 > r \geq 1.00$	0.15
$1.00 > r$	0.10

TABLE III: Value of p depending on $r = \frac{\sigma_{new}}{\sigma_{old}}$. Look-up table for algorithm1 (ALED)

However, the threshold does not change on detection of a speech frame. Hence threshold goes on decreasing as more number of low energy noise frames add to the noise frame buffer. Instances were observed where the threshold reached a value lower than most of/all the subsequent non-speech frames and then a large chunk of the recording ended up getting misclassified as speech. As a correction measure for this, we modified the original algorithm to keep a tab on the length of segment getting classified as speech. For the story-reading task, the continuous speech was observed to be not more than 2.5 seconds; so if the detected speech segment becomes longer than this we reset the threshold, and resume the algorithm from beginning of the on-going speech segment. Algorithm 1 provides the details of modified ALED.

Algorithm 1 ALED Algorithm

```
Initialize $k$, $speechLengthThreshold$, $E_{th,0}$, $winSize$, $bufferSize$, $fixedLengthBuffer$, $j$, $prevVar$
while $j < numberOfFrames$ do
    if $E_j > kE_{th,j}$ then
        $decision[j] \leftarrow 1$
        $length \leftarrow length + 1$
        if $length = 1$ then
            $tap \leftarrow j - 1$
        if $length > speechLengthThreshold$ then
            $j \leftarrow tap$ #Pointer for threshold reset check
            $E_{th} \leftarrow hardThreshold$ #Resetting Threshold
    else
        $length \leftarrow 0$
        $decision[j] \leftarrow 1$
        update $fixedLengthBuffer$
        $var \leftarrow$ variance $(buffer)$
        $ratio \leftarrow \frac{var}{prevVar}$
        $p \leftarrow$ look-up-table $(ratio)$
        $E_{th,j+1} \leftarrow (1-p).E_{th,j} + p.E_{silence,j}$
        $prevVar \leftarrow var$
        if $E_{th,j+1} > hardThreshold$ then
            $hardThreshold \leftarrow E_{th,j+1}$
    $j \leftarrow j + 1$
```

For a proper threshold, the speech segments seldom get misclassified as silence/non-speech but at the cost of too many false alarms. The trade-off between misdetection rate and false alarm rate can be clearly observed from the DET(Detection Error Tradeoff) curve in figure 1 for k (threshold scale parameter) as varying parameter. We note a 20% equal error rate (EER).

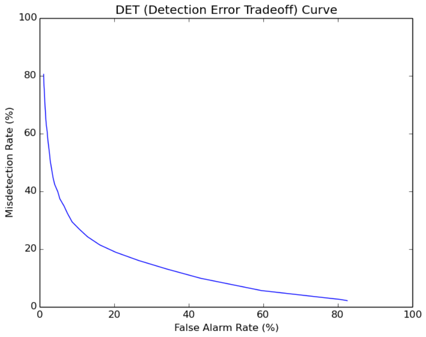

Fig. 1: Detection Error Tradeoff Curve for ALED

B. Zero Frequency Filtering (ZFF)

Another conventional feature widely used in VADs is harmonicity. We use zero frequency filtered signal[18] of the given signal. Pitch is the outcome of impulse train from source and its effect is spread over all frequencies. Zero frequency filtering helps efficient extraction of this information neglecting the effect of formants. The zero frequency filtered signal is obtained by passing the original speech signal through a cascade of two 0-Hz resonators followed by mean subtraction. The 0-hz resonator can be considered equivalent to the double integration of the signal. The output of the 0-Hz resonator is exponentially growing or decaying, making it difficult to capture discontinuities caused by pitch impulses. So, mean subtraction needs to be performed at the frame level, where average DC value of a frame is subtracted from all the samples of that frame.

The frame-level normalized first order correlation coefficient of the ZFF signal is found to be high in the voiced speech regions of foreground speech as compared to other regions. This value is then compared against a fixed threshold, as the ZFF signal is almost constant at the voiced regions as can be seen in figure 2. A detailed discussion of results has been done in section V. ZFF is a harmonicity based method and except for background talker, generator, and bell, almost all noise types are weakly harmonic as compared to voiced speech. So, it is less likely that a non-speech frame will get misclassified as speech.

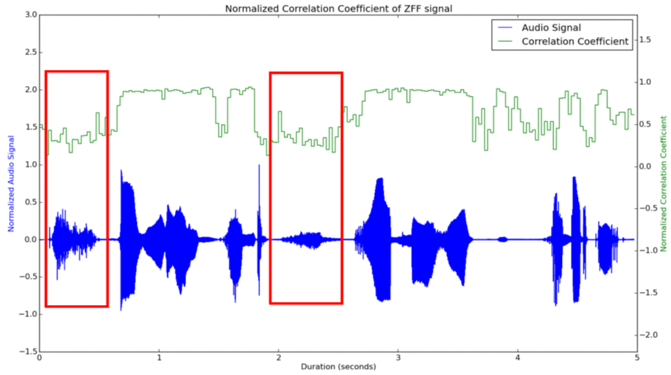

Fig. 2: Normalized First Order Autocorrelation Coefficient of ZFFS. Red boxes denote the regions of BR (non-harmonic), thus showing a drop in the corresponding correlation plot.

C. Proposed Two Pass System

1) Temporal Smoothing: The decision output of these two methods was observed to have a large number of very small silence ($\leq$ 200 ms) and speech spurts ($\leq$ 100 ms), introduced because of the sensitivity of the algorithms. Temporal smoothing (TS) was implemented in order to deal with such occurrences. We do an online updating of the raw decision output by classifying a small pause as its neighboring decisions. Taking all these points into consideration, a temporal smoothing algorithm (discussed in algorithm 2) was developed. It was individually applied on the ALED and ZFF decision outputs and improvement was observed in terms of both the misdetection and false alarm. In the algorithm 2, ALED or ZFF decision is referred to as 'rawDecision'.

1) 'segment' refers to an alternating 0,1 array, where all the frame-level contiguous zeros(ones) are mapped to a single zero(one),
2) 'noFrames' refers to an array having time duration of each segment in milliseconds,
3) 'update' function modifies the above two arrays after every iteration so that 'segment' retains its alternating 0,1 property.

Algorithm 2 Temporal Smoothing Function

```
function TemporalSmoothing(rawDecision)
    $j = 0$
    $[segment, duration] = \text{convertFtoS}(rawDecision)$
    $totalSegments = \text{length}(segment)$
    while $j < totalSegments$ do
        $N = duration[j]$
        if $(segment[j] = 1$ and $N \leq 100)$ then
            $segment[j] \leftarrow 0$
            $\text{update}(segment, duration)$
        if $(segment[j] = 0$ and $N \leq 200)$ then
            $segment[j] \leftarrow 1$
            $\text{update}(segment, duration)$
        $totalSegments = \text{length}(segment)$
        $j \leftarrow j + 1$
    $decision = \text{convertStoF}(segment, duration)$
    return $decision$
```

2) Hangover: Output of the above two methods after temporal smoothing was individually observed as Praat text grid along with audio. Most of the speech part that ALED had missed constituted undetected trailing speech edges at the start and end of detected speech segments. These regions were missed by ZFF approach too. One way to handle this is using hangover to delay the transition from speech state to noise state and vice versa.

A very naive way to do this is to declare some part (100ms in our case) before and after the detected speech segment to be speech. If the decision originally has high false alarms, then implementing hangover will worsen false alarms by a large amount. So, it is better to implement hangover scheme on the decision which has low false alarms, which would be ZFF in our case. Hangover was implemented on ZFFS after temporal smoothing as it reduced false alarms, and lesser the false alarms in the original decision better will be the results after adding hangover extensions.

3) Decision Logic: Motivated by the above, a higher misdetection rate for ALED was allowed so that we get reduced false alarm rate and the lost speech frames in the process can be recovered from (ZFF+TS+hangover) scheme. In a nutshell, we used the frame-level decisions obtained independently by (ALED+TS) and (ZFF+TS+hangover) and then logically ORed these two decisions to get the final decision, considering logic '1' as speech and logic '0' as non-speech. Improvements observed have been discussed in the next section in detail.

V. Results and Discussion

We have used standard AMR2 VAD as a benchmark for comparing our results. The source code from standard speech codec[19] was used for validation of our audio files. Frame-level misdetection rate and false alarm rate have been used as an evaluation metric [9][10] and the weighted average across the duration of all the audio files has been reported.

- Misdetection rate = $\frac{\text{Number of speech frames detected as noise}}{\text{Total number of speech frames in ground truth}}$
- False alarm rate = $\frac{\text{Number of noise frames detected as speech}}{\text{Total number of noise frames in ground truth}}$

Summarizing the pros and cons of ZFF and ALED in terms of error metric:

- Accuracy of ALED output depends on the value of k used; we can aim for either low misdetection rate or low false alarm rate at the cost of other depending on our application.
- ZFF approach gives a low false alarm for detected speech frames and so chances of any of non-speech frames getting misclassified as speech are very less. Thus, we can rely on ZFF-classified on speech segments.

Table IV presents the results for ALED, ZFF and the proposed two-pass algorithm discussed in implementation part. It also includes the benchmark obtained via AMR method. Low misdetection and false alarm rates are desired, but for comparison with the benchmark we measure the false alarm rate of the proposed method at misdetection rate of AMR (7%). So, in the case of modified ALED, the tunable parameter k was set such that the desired misdetection rate is achieved for false alarm rate comparison.

Fig. 3 presents a 10-second excerpt from one of the sound files in the dataset as observed in Praat software. The green curve stands for the intensity (dB) and the blue one for pitch (Hz) obtained using Praat. Except for tier 1, the tiers represent the decision outputs of all the methods listed in table IV and are numbered in the same order. Tier 1 intervals represent the manually labeled ground truth decisions. Speech segments are annotated using numerals, while non-speech segments use noise type name in tier 1 and 'SIL' in tier 2 to 7.

Sr. No.	Method	Mis-detection rate	False alarm rate
1	**AMR**	**7 %**	**39 %**
2	Modified ALED	22 %	18 %
3	Modified ALED + TS	13 %	15 %
4	ZFF	40 %	7 %
5	ZFF + TS	29 %	5 %
6	ZFF + TS + Hangover	16 %	9 %
7	**Two-pass**	**7 %**	**22 %**

TABLE IV: Evaluation of proposed and benchmark algorithms on our dataset

Following implications can be made from the table IV and figure 3:

1) How does two-pass help over ALED?
   (interval 14 of tier 1 - yellow box)
   - Tier 2 shows that some speech segments got misdetected by ALED because of intensity variation within the speech utterance.
   - Temporal smoothing does help remove very short duration misdetections (evident as we go from tier 2 to 3 for ALED and from tier 4 to 5 for ZFF) but still leaves out some longer misdetected segments.
   - The final two pass output (tier 7) is better at detecting speech segment than original ALED as well as ZFF.
2) How does two-pass help over ZFF?
   (interval 15 of tier 1 - red box)
   - From the previous points it might seem that using just method 6 should be sufficient. But it can be seen in tier 6 that this method alone cannot always extract all speech segments. It has been observed to give a false alarm rate of 40% at the benchmark misdetection rate of 7%.
   - If we had directly taken tier 6 decision here, the unvoiced sound in the middle of this interval would not have got detected as speech; thus considering ALED-detected speech frames as well helped in the two pass.
3) Drawback of the proposed system
   (intervals next to interval 12 and 13 of tier 1 - magenta boxes)
   - Here, we can see instances where ZFF (tier 4) has detected the non-speech part correctly (as also explained by its small false alarm rate), and ALED did not. But because of decision ORing logic, non-speech segments got misclassified as speech.
   - This is one of the major pitfalls of the proposed method.

Thus, summarizing the observations in terms of error metrics,

- Temporal smoothing helps reduce misdetections as well as false alarms in case of both ALED and ZFF, thus giving an overall improvement.
- Two pass system (method 7) gives 6% improvement in misdetection rate over (ALED+TS) (method 3) justifying that most of the frames missed by ALED were in fact near starting and trailing edges of detected speech segments.
- Same is valid for ZFF as well and is clear from the misdetection improvement when the hangover is applied to ZFF (from method 5 to method 6).
- Increase in false alarm from method 5 to method 6 can be attributed to the presence of non-zero false alarm rate in the former; hangover adds frames to falsely detected speech frames as well, thus further increasing the false detection.

On further analysis of sources of misdetection and false alarms, the following observations were made when method 3 was compared with method 7:

- Major improvement in misdetection was observed in files with heavy rain noise. This can be explained as the energy of heavy rain would be higher than the energy of trailing edges, and hangover helps recover the lost frames which are in fact speech.
- False alarm degrades largely because of background talker cases. This is clear from the fact that ZFF will show high periodicity for distinct background speech, thus adding to the false alarms in two pass as well.

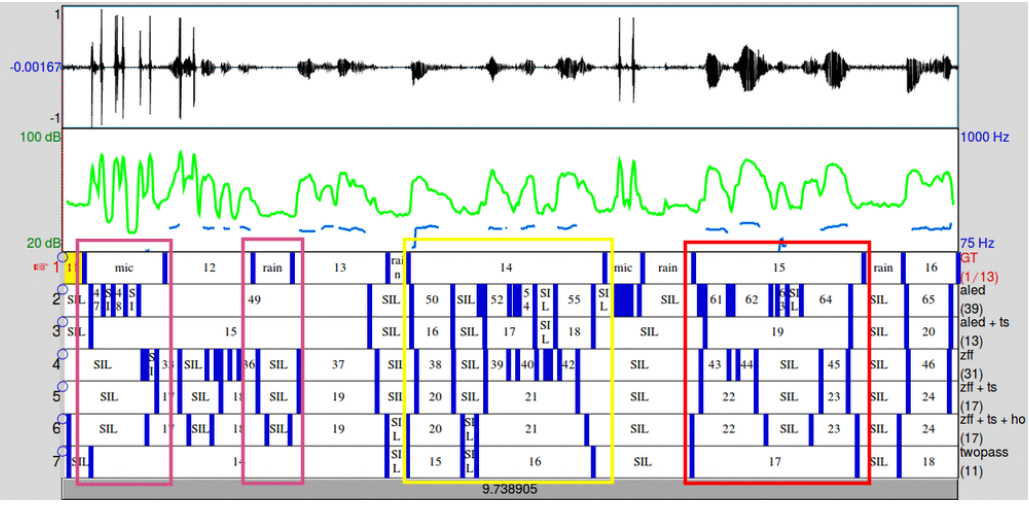

Fig. 3: Objective analysis and comparison of different methods; topmost tier is the ground truth

VI. Conclusion and Future Work

Our aim is to pre-process noisy speech before giving it to the ASR engine so that it does not have to go through extra modeling and adaptation schemes to accommodate for noise-induced mismatch and can focus on already challenging tasks of handling accented speech, mispronunciations, and disfluencies. Though we have been able to achieve an improvement of about 17% over the standard AMR, we require yet lower misdetections and false alarms for it to make an improvement in terms of ASR accuracy. Moreover, even if we are able to successfully classify speech and non-speech, segments with overlapping noise will still be an issue for the ASR. Either speech enhancement as a pre-processing step or noise-robust ASR techniques would be a possible way to tackle this problem.

Stationary noise occurrences of rain, generator and time-localised short duration noises like mic-type sounds, were reliably detected. However, background talker, school noise have been the most challenging noise types. We look forward to using better thresholding and hangover schemes for ZFF to enhance the performance. Also, as the algorithm is validated on a small dataset, the results might not be scalable. With more labeled data we will move over to classifier-based methods with additional features incorporated along with present features. Spectral shape, modulation frequency, signal variability are some of the features which look promising and will be tried out. We can also explore the advantage of using similar features in multiple spectral bands.

References

[1] "ASER: The Annual Status of Education Report (rural)," http://img.asercentre.org/docs/Publications/ASER\%20Reports/ASER_2012/fullaser2012report.pdf, ASER Centre, 2012.

[2] "Sensibol reading tutor app (2016)," http://sensibol.com/readingtutor.html, SensiBol Audio Technologies Pvt. Ltd.

[3] "LETS : Learn English Through Stories (2016)," http://www.tatacentre.iitb.ac.in/15mobitech.php, Tata Centre for Design and Technology at IIT Bombay.

[4] P. Rao, P. Swarup, A. Pasad, H. Tulsiani, and G. G. Das, "Automatic assessment of reading with speech recognition technology," in *Proc. of the 24th Int. Conf. on Computers in Education*, 2016.

[5] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[6] S. Dixit and D. M. Y. Mulge, "Review on speech enhancement techniques," *Int. Journal of Computer Science and Mobile Computing, IJCSMC*, vol. 3, no. 8, pp. 285–290, 2014.

[7] J. Jankowski Jr, "A new digital voice-activated switch," *COMSAT Technical Review*, vol. 6, pp. 159–178, 1976.

[8] J. Haigh and J. Mason, "Robust voice activity detection using cepstral features," in *Proc. of TENCON'93. Computer, Communication, Control and Power Engineering. IEEE Region 10 Conf. on*, vol. 3, 1993, pp. 321–324.

[9] T. V. Pham, M. Stark, and E. Rank, "Performance analysis of wavelet subband based voice activity detection in cocktail party environment," in *Proc. of The 2010 Int. Conf. on Advanced Technologies for Communications*, 2010, pp. 85–88.

[10] P. K. Ghosh, A. Tsiartas, and S. Narayanan, "Robust voice activity detection using long-term signal variability," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 600–613, 2011.

[11] A. Tsiartas, T. Chaspari, N. Katsamanis, P. K. Ghosh, M. Li, M. Van Segbroeck, A. Potamianos, and S. Narayanan, "Multi-band long-term signal variability features for robust voice activity detection." in *in INTERSPEECH*, 2013, pp. 718–722.

[12] M. Van Segbroeck, A. Tsiartas, and S. Narayanan, "A robust frontend for vad: exploiting contextual, discriminative and spectral cues of human voice." in *in INTERSPEECH*, 2013, pp. 704–708.

[13] J. Kola, C. Espy-Wilson, and T. Pruthi, "Voice activity detection," *Merit Bien*, pp. 1–6, 2011.

[14] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.

[15] ITU, "Silence compression scheme for g. 729 optimized for terminals conforming to recommendation v. 70," *ITU-T Recommendation G*, vol. 729, 1996.

[16] ETSI, "For adaptive multi-rate (amr) speech traffic channels," *ETSI EN*, vol. 301, p. 708, 1999.

[17] K. Sakhnov, E. Verteletskaya, and B. Simak, "Dynamical energy-based speech/silence detector for speech enhancement applications," in *Proc. of the World Congress on Engineering*, vol. 1, 2009, p. 801.

[18] K. S. R. Murty and B. Yegnanarayana, "Epoch extraction from speech signals," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602–1613, 2008.

[19] "Ansi-c code for the floating-point adaptive multi-rate (amr) speech codec," https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1400, version 13.0.0 (2015).