Decision fusion of voice activity detectors

Zaur Nasibov

Supervised by
Dr. Tomi Kinnunen

Master's thesis April 16, 2012
School of computing
University of Eastern Finland

Abstract

Voice activity detector (VAD) is an important part of any speech processing system. It is used to locate human speech segments in a given sound signal. The basic output of a VAD is *speech* or *non-speech* decision for every short segment of the given signal. Although the complexity of VAD algorithms varies from very simple to very complex ones, a simple algorithm can outperform robust VADs in particular noise conditions. Intuitively, combining the best properties of different VADs should lead to performance growth in a wide range of noise conditions. In this thesis we develop a concept of *VAD fusion* in which several VADs' outputs are combined in order to get a more accurate binary speech/non-speech classification of an input signal. The proposed fusion methods include majority voting, analysis of temporal context and simple trained model-based fusion. The base evaluation of standalone VADs and the fusion methods is carried on Aurora 2 corpus with more than 18 hours of data. Additional evaluation on three different corpora is carried out to confirm the results. The results indicate that the majority voting method can be used to achieve a different from standalone VADs classification behaviour with 1-2% improvement and the VAD fusion method based on preliminary trained speech model improves VAD performance by 5%. Our goal, in which we succeeded, was to study the possibility of improving VADs' results without interfering with the original algorithms but rather by combining their output.

Keywords:

voice activity detection, VAD, fusion,

Preface

Dear reader.
It took me almost three years to write this master thesis. One can say, “a long period...”. A long period indeed! I believe that this work will have a big impact on my life (and it already does) with the knowledge and new skills that came along. So here it is, thanks to support and contribution of my supervisor Dr. Tomi Kinnunen, who has been very patient with me and very punctilious with every idea, every sentence, every word that this thesis consists of. Many thanks to the management of Blancco OY, to Dr. Simo Juvaste and once more to Tomi, who allowed me to work full-time in parallel with writing my thesis. Finally, it could not be done without the support of my dearest friends and my beloved family.

Contents

List of abbreviations

AFE	Advanced Front-End
AMR	Adaptive Multi-Rate
ANN	Artificial Neural Network
ARMA	Auto Regressive Moving Average
ASR	Automatic speech recognition
bps	bits per sample
CS-ACELP	Conjugate Structure - Algebraic Code Excited Linear Prediction
DSR	Distributed Speech Recognition
DSR	Distributed Speech Recognition
ETSI	European Telecommunications Standards Institute
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
ITU	International Telecommunication Union

LFS	Line Spectral Frequencies
LTSD	Long-term Spectral Divergence
LTSE	Long-Term Spectral Envelope
MFCC	Mel Frequency Cepstral Coefficients
MULSE	Multiplication of Upper and Lower Signal Envelope
NIST	National institute of standards and technology
SAD	Speech activity detection
SIPU	Sound and Image Processing Unit
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
VAD	Voice activity detection
VoIP	Voice Over Internet Protocol
XAFE	Extended Advanced Front-End

Chapter 1

Introduction

1.1 Overview of Voice Activity Detection

Voice activity detection (VAD), also known as *speech activity detection* (SAD), is a technique used in speech processing in which the presence or absence of human speech in a sound signal is detected [30]. Voice activity detection plays an integral role in different speech signal processing systems such as in speech coding for cellular or IP phones and in front-end processing for recognition applications. It is also used as a part in various speech enhancement techniques like *noise reduction* and *echo cancellation* [14, 10].

A good example of voice activity detection application in modern cellular systems is *selective power-reserving transmission* [10]. For example, a VAD module can double the capacity of a GSM-based communication system by transmitting only the parts of a signal in which speech is present. This also leads to smaller battery power consumption [12].

While some VAD algorithms are considered to be generic and all-purpose, a demand for a VAD with a high performance in a specific environment remains. For instance, in Formula 1 driver-to-paddock[1] radio communication, which happens in an extreme loud engine noise conditions, requires a robust VAD for noise cancellation. Other challenges are found from *forensics* where the police wiretaps the suspects for several days and the VAD is then used to find the speech regions on “tape”. This is a tough VAD challenge, because the signal-to-noise ratio of the wiretap recordings can be very low [42, 15].

1.2 Challenges in voice activity detector design

Even a very simple VAD algorithms may have good performance when the input signal is clean and the speech is well-audible. Some heuristics might be required to correctly detect hissing and whistling sounds. As the power of the background noise increases, a VAD starts facing various challenges such as:

- **Low signal-to-noise ratio** (SNR). SNR is a generic measure of how much a signal has been corrupted by noise [41]. SNR is defined as a ratio of signal power to the noise power and will be discussed in detail in Section 2.2.1. A VAD has to detect speech correctly even if the background noise is very loud or a speaker is talking quietly [3]. This challenge is the hardest to deal in practice.
- **Rapid background noise variation**. Adapting to non-stationary background noise, e.g. on a construction yard, with loud equipment noise starting and stopping in a random order [36].

[1] An area at an automobile racecourse where the racing cars are parked and from which team engineers communicate with the drivers

- **Independence of language, accent and voice type**. A VAD has to have the same performance processing e.g. female Spanish contralto and men's Italian baritone.

1.3 Components of voice activity detector

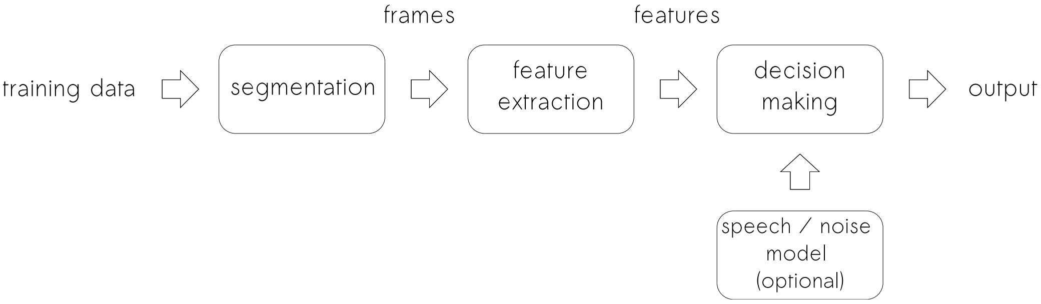


Figure 1.1: The components of VAD

Speech signal: The input of a VAD system is a digitized speech signal with some sampling rate. The IEEE defines a *voice-band channel* as "a channel that is suitable for transmission of speech or analog data and has the maximum usable frequency of 300 to 3400 Hz." [13]. VAD applications have been extensively used in digital phone systems [40, 12, 14], where the common sampling rate is $f_s = 8$ kHz. This implies maximum digital frequency of 4kHz.

Segmentation: The digital signal is processed in short-term *frames* of equal duration which is typically 10-30 ms in speech processing applications. This period is long enough to collect necessary data for further processing, yet short enough for the speech signal to remain stationary [29]. The frames are generally overlapped with frame advance equal to half or one third of the frame duration [26, 28].

Feature extraction: Since the raw input audio data is largely redundant and noisy for processing, feature extraction techniques are used to get the essential information about the data that would be enough for further processing. The goal of feature extractor is to compress every frame by mapping its data to a vector of features so that the number of features $\ll$ the number of samples in a frame. The features carry the information that should be enough for a VAD to classify the frame.

Some of the features that have been proposed for VAD include *zero crossing rate* [3], *full-band* and *low-band energy* [3], *multiplication of upper and lower signal envelope* [11], *spectral entropy* [34], *long-term spectral divergence* [31] *and mel frequency cepstral coefficients* (MFCCs) [17].

Decision making: At this step a VAD classifies each frame as either speech or non-speech (noise). The decision making rule might use simple (fixed threshold-based) as well as very complex (support vector machine (SVM), hidden Markov model (HMM)-based) classifiers to produce the output.

1.4 VAD types

A vast amount of different VAD algorithms have been developed. Their complexity varies from very simple to very complex ones. The major industrial VAD algorithms that have been standardised include G.729, adaptive multi-rate (AMR), advanced front-end (AFE) and Skype SILK which are considered to be generic and all-purpose [3, 37, 36]. The common part of these algorithms is the run-time background noise estimation for which an additional, potentially simpler VAD [37, 3], or features extracted from previous frames [36] are used (Fig. 1.2). An adaptive threshold is calculated from the noisy parts of the signal, which is then used to estimate whether the frame contains speech.

Most of the industrial VADs use a threshold applied on feature vectors extracted from the signal. If the measured parameters exceed the threshold, then a frame is declared as speech [6]. The thresholds can be fixed and determined initially, for example, by using genetic optimization [11]. Alternatively, they might be adaptive and depend on processed signal frame features [8].

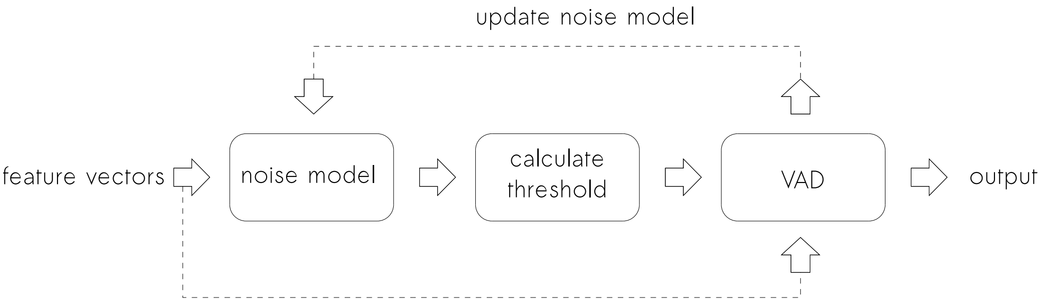


Figure 1.2: VAD with run-time noise estimation

Unlike a noise-adaptive VAD, a data-driven VAD requires previously trained model. Various approaches to data-driven VADs include but are not limited to: *hidden Markov models* [35, 5], *Gaussian mixture models* [22], *support vector machine* [17] and *artificial neural network* [11].

1.5 Motivation of research

Every VAD algorithm has advantages and disadvantages. A VAD can perform very good in high sound-to-noise conditions, but fail under low SNR conditions. Another VAD may perform worse in high SNR environment, but catch up and outperform the first VAD in low SNR conditions. How can we combine the best properties of different VADs to produce a better result than a VAD does as a standalone algorithm? One possible way is to combine the algorithms and create a new VAD by *data fusion* or *classifier fusion.*

In this thesis by *VAD fusion* we understand a technique in which several VAD outputs are combined in order to get a more accurate speech/non-speech classification of a signal (Fig. 1.3).

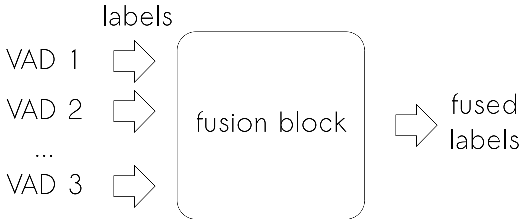


Figure 1.3: VAD fusion scheme. Multiple speech/non-speech input labels are fused into a single output label.

VAD fusion is a particular case of combining classifiers' decisions. The classifiers fusion is a next step to be taken, when a large amount of various performance-competing classifiers are available. This topic has received a lot of attention recently [20]. The two main strategies in combining classifiers are *fusion* and *selection.*

- **Fusion**: each classifier is applied to the input data. A fusion scheme is applied to the output of the classifiers in order to produce final decision.
- **Selection**: each classifier is applied to a particular subset of input data, and is responsible for classifying the objects from that subset. The system can choose the classifier that outperforms other classifiers

on that kind of data. This could be achieved by prior knowledge of the input data properties for example in forensics.

- Combination of fusion and selection lies somewhere in-between the two said techniques. For example, several classifiers are responsible for a subset of the input data and a fusion scheme is applied to the output results.

The goals of this thesis are to, firstly study and compare standalone VADs behaviour in different noise environments, secondly research and study feasibility of VAD fusion and finally introduce several new methods of VAD fusion.

Chapter 2

Voice activity detection methods

2.1 Speech signal processing in VAD applications

Speech signals can be analyzed either in the time domain or in the frequency domain. Thus, the processing methods that involve the waveform of the speech signal directly are called *time-domain* methods. In contrast, *frequency-domain* methods involve (either explicitly or implicitly) some spectral representation. An example of a time-domain waveform and the spectrum of the same segment is shown in Fig. 2.1. [16, 29].

2.1.1 Short-time processing

One of the most widely used speech signal processing techniques is short-time processing. The short-time processing techniques are based on assumption that the properties of the speech signal change relatively slowly with time.

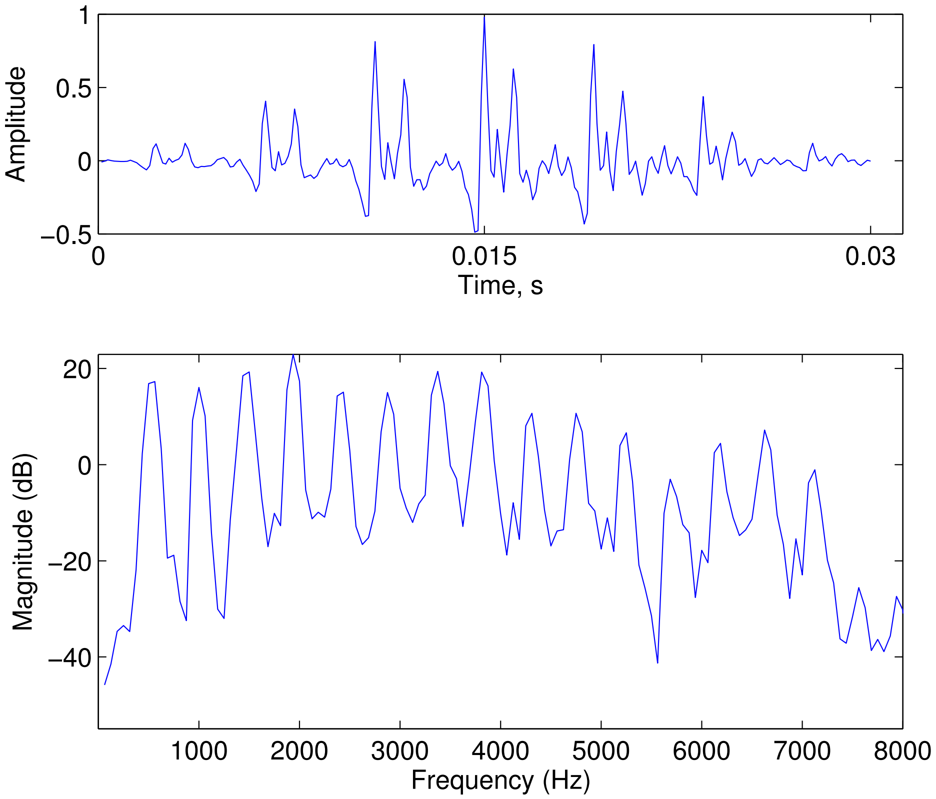


Figure 2.1: And example of a waveform and corresponding magnitude spectrum of signal frame. The first plot shows a 30-ms frame of a sound signal. The second plot represents the magnitude spectrum of the signal. It is obtained by squaring the absolute values of 512-point discrete Fourier transform of the signal.

The signal is divided into short-term *frames* which often overlap one another and the frames are processed individually.

Most of the short-time processing techniques can be represented mathematically in the form [29]:

$$Q_n = \sum_{m=-\infty}^{\infty} T[s(m)]w(n-m) \tag{2.1}$$

Where $s(m)$ is a speech signal, $T[\ ]$ is a linear or non-linear signal transformation and w is a window function (see Section 2.1.3).

2.1.2 Long-term processing

Generally a VAD algorithm is intented to work in a runtime environment, producing decisions based on current and previous frames [3, 37, 36]. Yet, there are tasks (e.g. forensics) in which a speech processing system is applied to a recorded signal. This makes long-term signal statistics available to a VAD, leading to increased speech detection robustness [31, 34, 11].

2.1.3 Window functions

The purpose of the windowing is to reduce the effect of the spectral artefacts that result from the framing process. According to the convolution theorem the $T \cdot w$ multiplication in (2.1) corresponds to convolution of the signal spectrum with the window function response. In other words, the transfer function of the window will be present in the observed spectrum [16]. The Hamming window is assumed to be most widely used for speech processing system [29, 28] and is defined as follows:

$$w(n) = \begin{cases} 0.54 - 0.46\ \cos\left(\frac{2\pi n}{N-1}\right), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Here, N is a number of samples in a frame.

2.1.4 Hangover

Some VAD algorithms might work inefficiently on the borders of speech fragment start and end points, or misclassify e.g. hissing speech sounds as noise. However, it is possible for a VAD to wait for several frames to be above or below a threshold level, before reporting the decision on the frame currently being processed. A set of empirical rules used to smooth the final VAD decision based on previously made decisions is united into *hangover* mechanism [3, 37, 36, 33]. For example, the hangover mechanism in G.729 VAD consists of four steps [3]:

1. The frame is marked as “speech” if the frame energy (defined below in Section 2.2.1) is above full-band energy difference.

2. The frame is marked as “speech” if two previous frames are also “speech”-marked frames, and the absolute energy difference between the current and previous frames is under a constant threshold ($N_2 = 10$). This extension is performed only for two consecutive frames.

3. A “non-speech” decision is extended to the frame if the previous 10 frames are also marked as “non-speech”, and the difference between the current and previous frames’ energy is under a constant threshold ($N_1 = 4$) and the previous frame is also marked as “speech”.

4. The “speech” decision is changed to “non-speech” if the current frame energy is below the noise floor by a constant threshold ($N_0 = 128$), the second reflection coefficient is smaller than 0.6, and the first or second smoothing step did not take place.

2.2 Time-domain features

2.2.1 Signal energy and signal to noise ratio

The energy (also known as power) of the n^{th} frame of a discrete signal $s(m)$ is a feature that reflects signal's amplitude variations [1]. It is defined as follows:

$$E_n = \frac{1}{N}\sum_{i=1}^{N} s_n^2(i). \tag{2.3}$$

Here, N is the number of samples per frame and s_n is the n^{th} frame.

In high signal-to-noise ratio environments, the energy of the lowest level speech sounds (e.g. weak fricatives) exceeds the background noise energy, and thus a simple energy measurement suffices as speech activity indicator (Fig. 2.2). However, such ideal recording conditions are not practical for most applications [29].

Signal to noise ratio Let P_s and P_n denote the average energy of speech and noise frames of signal s. Then the signal-to-noise ratio of s is calculated as follows [41]:

$$SNR = 10\log(\frac{P_s}{P_n}) \tag{2.4}$$

2.2.2 Alternative features

Zero-crossing rate In the context of discrete-time signals, a *zero-crossing* is said to occur if successive samples have different algebraic signs. The rate

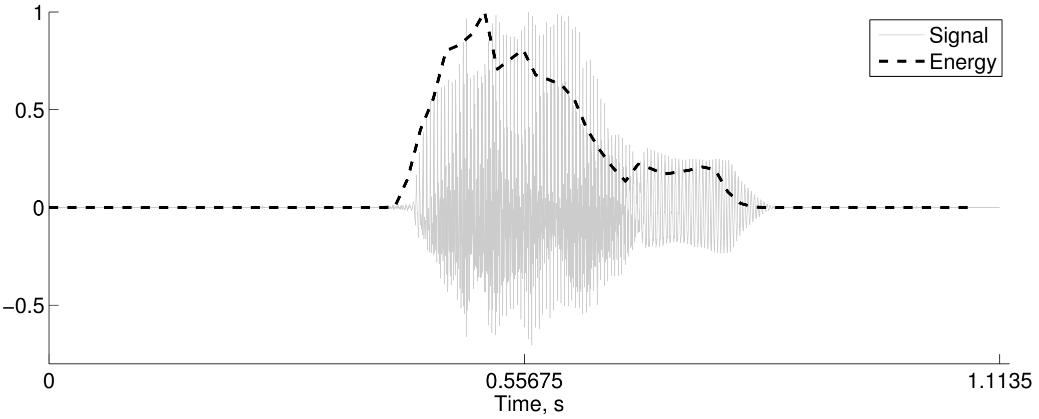


Figure 2.2: Speech signal energy curve. The above signal contains the words "zero, eight" spoken by a female voice. High energy indicates a speech presence in a frame. The energy drops significantly at the end of the word "eight", due to a dull sound ending

at which zero crossings occur is a simple measure of the frequency content of a signal (Fig. 2.3).

For a given frame of N samples, zero-crossing rate is defined as follows:

$$Z_n = \frac{1}{2N}\sum_{i=1}^{N}(|\text{sgn}(s_n(i)) - \text{sgn}(s_n(i-1))|) \tag{2.5}$$

The model of speech production suggests that the energy of speech is concentrated below 3 kHz because of the spectrum fall-off introduced by the glottal wave, whereas for noise, most of the energy is found at higher frequencies. In practice it means, that a frame contains speech if the zero-crossing rate is low [29].

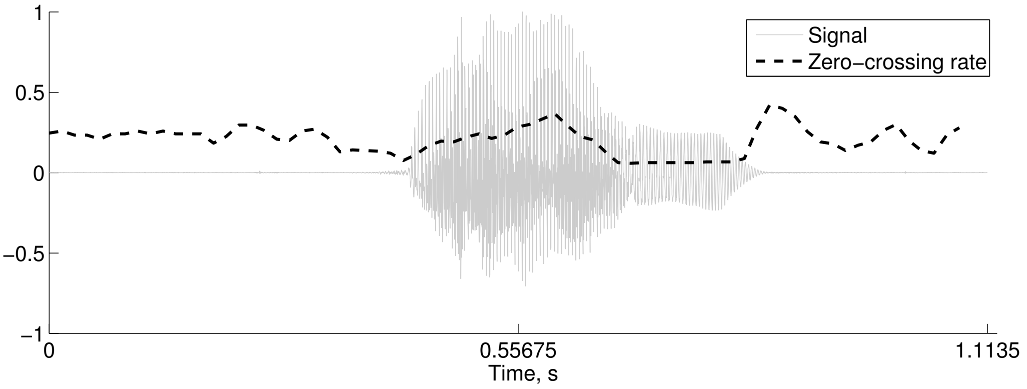


Figure 2.3: Zero-crossing rate of a speech signal. In theory, zero-crossing rate should be lower in speech region, but as it is seen on this plot, the statement is not always true.

Signal envelope For a given frame of N samples, MULSE is defined as follows:

$$M_n = |max(s_n(i)) \cdot min(s_n(i))| \quad | \; i \in (1..N) \tag{2.6}$$

MULSE is a time-domain feature calculated by multiplying upper and lower parts of signal envelope (Fig. 2.4) [11].

2.3 Spectral-domain features

2.3.1 Entropy

When the SNR of a signal is very low (e.g. smaller than 0 dB), time domain processing is difficult, since the features' values of speech and non-speech frames do not differ as much as they do with high SNR.

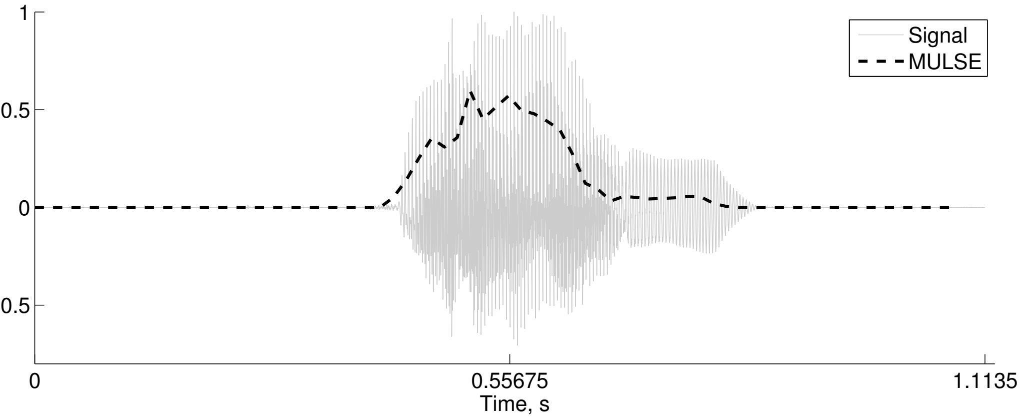


Figure 2.4: MULSE curve of a speech signal. MULSE is high in frames that contain speech. Like frame energy, MULSE drops significantly on hissing ending of the word “eight”

Information entropy is a probability measure of information contained in a message [38]. The application of the concept of entropy to the speech detection problem is based on the assumption that the signal spectrum is more organized during the speech frames than during nonspeech frames. Let $s(m)$ be a discrete speech signal divided into overlapping frames and let $S_n(f)$ - denote the magnitude spectrum of the n^{th} frame for frequency bin f. The measure of entropy is defined in the spectral energy domain as follows [34]:

$$H(|S_n(f)|^2) = -\sum_{f=1}^{\Omega} P(|S_n(f)|^2) \cdot \ln(P(|S_n(f)|^2)) \tag{2.7}$$

where

$$P(|S_n(f)|^2) = \frac{|S_n(f)|^2}{\sum_{k=1}^{\Omega} |S_n(k)|)}$$

is the probability of f^{th} band magnitude spectrum in frame k. It is called *probability mass function* (pmf) and defines the probability of a discrete-random variable X taking a value of x_i, $P(X = x_i)$ [44].

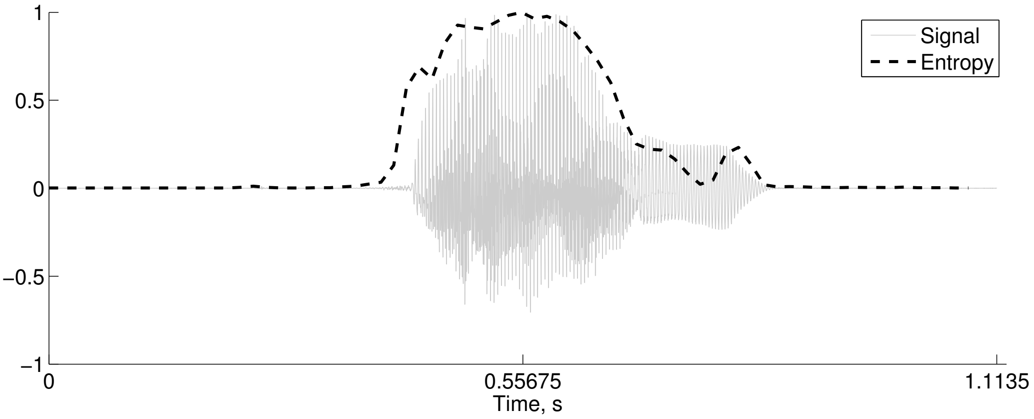


Figure 2.5: Spectral entropy of a speech signal. High entropy value indicates speech presence in a frame. From these informal visual inspections, entropy outperforms other measures in detecting dull sounds, e.g. the last entropy peak on the plot highlights “t” sound from word “eight”

A long-term information can be applied to make the resulting spectrum less dependent of speech type: the spectrum of each frame is divided by average spectrum computed over all frames [34].

2.3.2 Alternative features

Long-term spectral divergence Let $s(n)$ be a discrete speech signal divided into overlapping frames and $S_n(f)$ - n^{th} frame magnitude spectrum for band f^{th}.

The Mth-order *long-term spectral envelope* (LTSE) denotes the maximum value of $S_j(f)$ in $j \in [n - M, n + M]$ temporal context. LTSE is defined as follows [33, 31]:

$$\text{LTSE}_M(f,n) = max\{S_j(f)\}_{j=n-M}^{j=n+M} \tag{2.8}$$

The N-order long-term spectral divergence between speech and noise is defined as the deviation of the LTSE with respect to average noise spectrum magnitude $S(f)$ for the f band, $f = 0, 1, \ldots, \Omega$ and is given by:

$$\text{LTSD}_M(n) = 10\log_{10}\left(\frac{1}{\Omega}\sum_{f=0}^{\Omega-1}\frac{\text{LTSE}_n^2(f)}{S_{noise}^2(f)}\right), \tag{2.9}$$

where S_{noise} is the mean noise spectrum estimated by averaging the noise spectrum magnitude during a short initialization period (e.g. from the first K frames, assumed to be non-speech).

2.4 Industrial VAD algorithms

2.4.1 G.729

The International Telecommunication Union (ITU) has adopted a toll-quality speech coding algorithm known as *conjugate structure - algebraic code excited linear prediction* (CS-ACELP). The corresponding recommendation is known as G.729. The Annex B recommendation describes a VAD algorithm that is used as a front-end in the G.729 codec family [3].

G.729 utilizes the following features to make voice activity decision:

- Line spectral frequencies (LFS) - a set of linear prediction coefficients is derived from the first 11 terms of the autocorrelation using G.729 (Annex A) procedures, which are then converted to a set of LFS.

- Full-band energy
- Low-band energy, measured at 0-1 kHz band
- Zero-crossing rate

The G.729 VAD works at 10-ms frame rate. The difference parameters are computed by subtracting the current frame's feature values from the running average of each feature. These variables form the points generated by frames of active voice are clustered in a certain region (hypervolume) of the four-dimensional space, while the points generated by frames of inactive voice are clustered in another region (the regions may overlap). A three-dimensional piecewise linear decision boundary identifies the inactive voice region, and its complement - the active voice region. Fourteen hyperplanes are used, each defining a section of the decision boundary. The parameters for each hyperplane were determined by visual inspection of the points' distribution over a large corpus, using scatter plots. Although the visual inspection method is the easiest to perform, it does not ensure the best performance at all.

Finally the VAD decision is smoothed to reflect the stationary nature of both the speech signal and the background noise. This smoothing and correction uses four steps of heuristic rules which resulted from extensive observations of the initial VAD decision [3].

2.4.2 Adaptive multi-rate (AMR)

Adaptive multi-rate (AMR) audio codec is a patented audio data compression scheme optimized for speech coding [25]. The European telecommunications standards institute (ETSI) standard EN 301 708 describes two voice activity detection algorithms adopted for AMR.

AMR VAD type I algorithm utilizes the following features for voice activity detection:

- **Filter bank** and 9 sub-band energy levels.
- **Pitch**. The purpose of the pitch detection function is to detect vowel sounds and other periodic signals.
- **Tone**. Tone detection is used to detect information tones (e.g. call progress tones, such as ringing tone or busy tone [2]), since the pitch detection function can not always detect these signals.

The AMR VAD also includes *correlated complex signal analysis*, which is used to detect correlated signals, such as music since the pitch and tone detection functions can not always detect these signals.

The intermediate VAD decision is made for every 20ms frame and is calculated based on the comparison of the background noise estimate and feature levels of the input frame. Finally, the VAD flag is calculated by adding hangover to the intermediate VAD decision.

The AMR VAD type II algorithm utilizes sub-band energy levels and SNR computed in spectral domain. The intermediate VAD decisions are made every 10ms, and the final decision is calculated for 20ms frame [36].

2.4.3 Advanced front-end (AFE)

The performance of speech recognition systems receiving speech that has been transmitted over mobile channels can be significantly degraded when compared to using an unmodified signal. ETSI AFE[1] codec was designed to perform as a part of a distributed speech recognition (DSR) system, in which an error protected data channel is used in parallel with the speech signal channel, to send a parametrized representation of the speech, which is suitable for recognition [37].

AFE includes two VADs and a voice classification functional block.

- **VADNest** is a noise estimation VAD, whose output is used for noise reduction via Wiener filtering procedure. VADnest operates on 10ms frame rate and utilizes logarithmic frame energy for voice activity detection.
- **VADVC** is a voicing classification VAD. VADVC utilizes channel frame computed per 23 mel filter-banks, a static threshold table and a hangover scheme for voice activity detection.
- Classification utilizes VADVC's output, frame energy, upper-band signal and pitch period estimate to classify a frame. The output is one of four voicing classes: non-speech, unvoiced, mixed-voiced, fully-voiced.

The output threshold used in this thesis for AFE VAD is set as following:

$$\text{AFE} = \begin{cases} 1 & \text{if output class} \in \{\text{mixed-voiced}, \text{fully-voiced}\} \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

[1] Typically, "AFE VAD" referrs to ETSI ES 202 212 standard's extended advanced front-end (XAFE) VAD algorithm. In this thesis, XAFE's VAD is also referred to AFE.

2.4.4 SILK

SILK is the speech codec for real-time, packet-based voice communications developed for popular Skype VoIP application. In SILK, the input signal is processed by a VAD to produce a measure of voice activity, and also spectral tilt and signal-to-noise estimates, for each frame. The VAD uses a sequence of half-band filterbanks to split the signal into four sub-bands: 0 - $f_s/16$; $f_s/16$ - $f_s/8$; $f_s/8$ - $f_s/4$; $f_s/4$ - $f_s/2$. Here f_s is the sampling frequency, which is either 8, 12, 16 or 24 kHz. The lowest sub-band, from 0 - $f_s/16$ is highpass filtered with a first-order moving average filter to reduce the energy at the lowest frequencies. For each frame, the signal energy per sub-band is computed. In each sub-band, a noise level estimator tracks the background noise level and an SNR value is computed as the logarithm of the ratio of energy to noise level. Using these intermediate variables, the following parameters are then calculated for use in VAD's pitch analysis and the other SILK modules [39]:

- **Speech activity level**, which is based on the average SNR and a weighted average of the sub-band energies.
- **Average SNR**. The average of the sub-band SNR values.
- **Smoothed sub-band SNRs**. Temporally smoothed sub-band SNR values.
- **Spectral tilt**. A weighted average of the sub-band SNRs, with positive weights for the low sub-bands and negative weights for the high sub-bands. The input signal is filtered by a highpass filter to remove the lowest part of the spectrum that contains little speech energy and may contain background noise. Finally, the signal is processed by the

open loop pitch estimator. Although SILK allows high-frequency input signal, the pitch analysis operates on signals downsampled to 4 and 8 kHz. This is done in order to reduce computational complexity.

Table 2.1: Summary of the attributes of different VADs

VAD	Energy, Entropy	G.729 [3]	AMR1 [36]	AMR2 [36]	AFE [37]	SILK [39]
Usage	General	VoIP	GSM, 3G-GSM, audio compression		audio compression and ASR	VoIP
Features	Energy and entropy	Full-band and low-band energies, ZCR and LFS	Sub-band energy, pitch and tone	Sub-band energy and SNR calculated in spectral domain	Sub-band energy and pitch	full-band and sub-band SNR, spectral tilt
Supported sampling frequencies	Any	8kHz	8 kHz		8, 11, 16 kHz	8, 12, 16, 24 kHz
Voice activity decision step	30 ms	10 ms	20 ms		30 ms	20 ms
Noise model/detection approach	Fixed threshold	Additional simplified VAD	Features from previous frames		Additional simplified VAD	Features from previous frames
Hangover mechanism	No	Yes				No
Output	binary				quaternary	binary

Chapter 3

Fusion of voice activity detectors

As the English proverb says, two heads are better than one. A single VAD may perform reasonably well in high SNR conditions but fail at low SNR. On the other hand, VAD may have a higher misclassification rate but work consistently across different SNRs. A team of VADs, in which every algorithm complements the others should perform better than every VAD per se.

The technique of binding several VADs in a team is called VAD *fusion*. In this section, we describe different VAD fusion techniques.

3.1 Measuring diversity of VAD algorithms

Intuitively, the VADs to be combined should be *diverse*. There is no advantage in combining VADs that behave the same way. Therefore, *diversity* (negative dependence, independence, orthogonality, complementarity) among the individual team members in a fusion pool has been recognised as a key issue for successful fusion [19].

Consider two VADs (VAD_1, VAD_2) running on a training data set and a 2x2 table that summarizes their output, as shown in Table 3.1. The entries in the table are the probabilities for the respective pair of correct/incorrect outputs.

Table 3.1: Two VADs relationship table

	VAD_1 correct	VAD_1 wrong
VAD_2 correct	a	b
VAD_2 wrong	c	d

Here, a, b, c and d are the probabilities for the respective pair of correct/incorrect outputs. $a + b + c + d = 1$

It is implied that a training data set contains *ground truth* (GT) labels, which are used for counting the occurrences of each binary vector for speech and non-speech classes.

Based on table 3.1, several diversity measures can be computed. For two VADs and their relation, *correlation coefficient* is defined as follows [19]:

$$\rho_{1,2} = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \tag{3.1}$$

Here, $\rho \in [-1, 1]$, $\rho = 1$ corresponds to VADs with absolutely similar output whereas $\rho = -1$ corresponds to VADs with totally different output. The ρ value is a particular case of *Pearson's correlation coefficient* ρ_ϕ defined for two binary variables [4].

The task of selecting K diverse VADs from V available VADs for a fusion, is the task of choosing VADs with minimal summary diversity distance between them. It can formalized as follows:

Table 3.2: Two VADs relationship example

Ground truth	0 0 1 1 1 1 0 0 0 0 1 1
VAD_1	0 0 0 1 1 1 1 0 0 0 0 1
VAD_2	0 0 1 1 1 0 0 0 0 0 0 0
Relation	*a a b a a c b a a a d c*

The relation is calculated by comparing VAD_1 and VAD_2 output labels with the ground truth labels. The comparison is done in a context of every frame: First, the VAD_1 output is compared to the ground truth, to determine if VAD_1 output is correct. Then, the VAD_2 output is also compared with the ground truth, do determine if VAD_2 output is correct. Finally, these values and Table 3.1 is used to identify the relationship between the VADs.

Let Υ_V define a set of all available VADs ($\Upsilon_V = \{\text{VAD}_1, \text{VAD}_2, \ldots\}$) and Υ_K define a set of all K-combinations of Υ_V. Formally, the K VAD combination of a set Υ_V is an unordered set of distinct K VADs from Υ_V. The number of K-combinations is determined as follows [24]:

$$|\Upsilon_K| = \binom{V}{K} = \frac{V!}{K!(V-K)!} \tag{3.2}$$

Let C_k define k^{th} VAD combination in Υ_K and let μ_{ij} define the chosen correlation metric value between VAD_i and VAD_j (such as the correlation (3.1)), where VAD_i, $\text{VAD}_j \in C_k$. The task of finding the least correlated K VADs out of V available VADs is a subject of selecting the VADs for which the following sum is minimized:

$$\sum_{C_k} \mu_{ij} \rightarrow \min \mid \forall i \neq j; \forall C_k \in \Upsilon_K \tag{3.3}$$

Although K could be determined experimentally, its value is fixed in the experiments carried out in this thesis ($K = 3$). The reason for fixing K is not related to calculating the correlation coefficients, but to the overall amount of VAD combinations. For example: consider a pool of six VADs. In this case, $K \in 2..6$. The overall amount of combinations is calculated by (3.2): $15 + 20 + 15 + 6 + 1 = 57$. This number is almost three times bigger than the amount of 3-VAD combinations: twenty VAD combinations out of six available VADs.

3.2 Fusion methods

3.2.1 Majority voting

Majority voting is the binary decision rule, which involves a group of voters and selects an alternative for which more than half votes were given [20]. Formally the majority voting rule is denoted by the following equation:

$$\Phi_n = \begin{cases} 1 & \text{if } \frac{1}{V}\sum_{i=1}^{V} \mathrm{vad}_i^n > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Here, n is the index of the current frame; Φ_n is the final decision made for the current frame; V is the number of VADs involved in a fusion; $\mathrm{vad}_i^n \in \{0, 1\}$ is the binary output of the i^{th} VAD for the current frame. Table 3.3 shows and example of majority voting VAD for $V = 3$.

In social choice theory, the so-called *May's theorem* states that simple majority vote is the only procedure which is *anonymous*, *dual*, and *monotonic* [23]. It means that a group decision is the simple majority decision if and

Table 3.3: Three VADs majority voting example

Frame	1	2	3	4	5
vad_1	1	1	1	0	1
vad_2	1	1	0	0	0
vad_3	1	0	1	0	0
Φ	1	1	1	0	0

only if each voter is treated equally, each alternative is treated equally, there is only one winner and if a voter changes the vote, it will still affect the end result as any other vote would do. May's theorem also implies that majority voting is true only when there is an odd number of voters and ties are not allowed.

A typical VAD algorithm meets all May's theorem requirements. AFE is an example of a VAD which violates the decisive rule, since it makes soft rather than hard decisions (see Section 2.4.3). However, a fixed threshold can be used forcing AFE to produce "speech" and "non-speech" labels.

3.2.2 Temporal context majority voting

Most of industrial VADs utilize a hangover scheme involving several previously made decisions, to compute the final decision for the current frame (see Section 2.1.4). The *temporal context majority voting* may be considered as a simple hangover scheme. Here, the fusion scheme utilizes V VADs outputs of d previous, current and d following frames:

$$\Phi_n = \begin{cases} 1 & \text{if } \frac{1}{V \cdot J} \sum_{j=n-d}^{n+d} \sum_{i=1}^{V} \text{vad}_i^j > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Table 3.4: Temporal context majority voting example

Frame	1	2	3	4	5
vad$_1$	1	**1**	**1**	**0**	1
vad$_2$	1	**1**	**0**	**0**	0
vad$_3$	1	**0**	**1**	**0**	0
Φ	1	1	**0**	0	0

The bold-coloured labels are involved in fusion calculation. Note that the fusion result differs from the majority voting result in table 3.3.

The boundary conditions, for which $n < d$ or $n > N - d$ (where N is the overall number of frames) have to be solved separately. In this thesis, a simple majority voting is used for the boundary cases, as illustrated in Table 3.4.

3.2.3 Histogram model-based fusion

The previously described simple and temporal context majority voting fusion methods have no preliminary knowledge of the input data. Intuitively an algorithm that has that kind of knowledge should show better performance. The excerpt of this knowledge is kept in a mathematical model. In this thesis we suggest using *histogram model*-based approach. Consider three VADs producing binary speech and non-speech labels. Every combined output frame X, is one of the $2^3 = 8$ possible output combinations:

$$X \in \{(1,1,1);\ (1,1,0); \ldots (0,0,0)\}.$$

Given the ground truth, it is possible to calculate the frequency of every

VAD combination output for speech and non-speech parts of the signal, as follows:

$$P(X|\text{speech}) = \frac{\text{\# of frames labeled as X}}{\text{\# of speech frames according to the GT}} \tag{3.6}$$

$$P(X|\text{nonspeech}) = \frac{\text{\# of frames labeled as X}}{\text{\# of nonspeech frames according to the GT}} \tag{3.7}$$

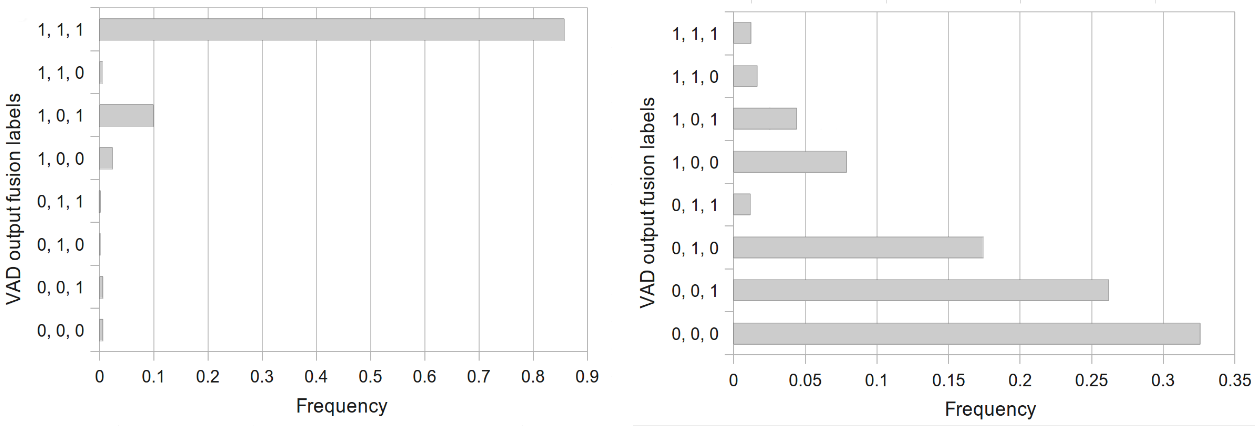
The overall probability of a VAD combination output occurrence is denoted as follows:

$$P(X) = \frac{\text{\# of frames labeled as X}}{\text{total \# of frames}}.$$

These calculations are carried out in training phase. Figure 3.1 shows an example of the histograms. On the right histogram we can see the frequencies of VAD combinations' outputs corresponding to "speech" ground truth labels. The left histogram shows the frequencies of VAD combinations outputs' corresponding to "non-speech" ground truth labels. We can learn a lot about the given VAD combination from these histograms and use that knowledge while performing runs on test data sets. For example, the 1,1,1 output combination most probably means that the frame contains speech. Vice-versa, the 0,0,0; 0,0,1 or 0,1,0 combinations probably mean that the frame does not contain speech.

In Bayesian decision theory, $P(X|\text{speech})$ and $P(X|\text{nonspeech})$ speech denote the *conditional probability* of X belonging to either speech or non-speech classes. This probability is called *likelihood* [45].

The well-known Bayes' theorem expresses *a posteriori* probability in terms



(a) Fusion frequencies of detecting speech

(b) Fusion frequencies of detecting non-speech

Figure 3.1: An example of VAD decision histograms for speech and non-speech ground truth

of *a priori* probability and conditional probability as follows [45]:

$$\begin{aligned} &P(\text{speech}|X) = \tfrac{P(X|\text{speech})P(\text{speech})}{P(X)} \mid \forall X, P(X) > 0. \\ &P(\text{speech}|X) + P(\text{nonspeech}|X) = 1 \end{aligned} \tag{3.8}$$

One of the important data properties that could be calculated using ground truth labels is the probability of speech frame occurring in the data:

$$P(\text{speech}) = \frac{\#\text{ of frames labeled as speech}}{\text{total }\#\text{of frames}}. \tag{3.9}$$

Accordingly, $P(\text{nonspeech}) = 1 - P(\text{speech})$ is considered as probability of non-speech frame occurring in the data. In Bayesian decision theory, these probabilities are called *a priori* probabilities [45].

In the test phase, the task of deciding whether a frame labeled as X is a speech or non-speech frame, is a task of comparing probabilities as follows:

$$P(\text{speech}|X) \geq P(\text{nonspeech}|X) \tag{3.10}$$

Here, $P(\text{speech}|X)$ and $P(\text{nonspeech}|X)$ are *a posteriori* probabilities of classifying X as speech or non-speech respectively.

Therefore, the decision rule (3.10) can be rewritten, by applying Bayes' theorem, as follows:

$$\frac{P(X|\text{speech})}{P(X|\text{nonspeech})} \geq \frac{P(\text{nonspeech})}{P(\text{speech})}$$

The $l_r(X) = \frac{P(X|\text{speech})}{P(X|\text{nonspeech})}$ ratio is called the *likelihood ratio* [45]. The $\Delta = \frac{P(\text{nonspeech})}{P(\text{speech})}$ ratio is non other than an inverse value of speech-to-nonspeech ratio (4.1), or a ratio of a number of non-speech to a number of speech frames computed by means of the training data ground truth.

In this thesis we make a simplified assumption that the shape of the histograms (conditional probabilities) and speech-to-nonspeech frames' lengths ratio match for training and test data.

Finally, the fusion decision rule based on the histogram model is defined as follows:

$$\Phi_n = \begin{cases} 1 & \text{if } l_r(X) \geq \Delta \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

Chapter 4

Experimental setup

4.1 Corpora for VAD evaluating

4.1.1 Aurora 2

The Aurora project was originally set up to establish a world wide standard for the feature extraction software which forms the core of the front-end of a *distributed speech recognition* (DSR) system.

Aurora 2 (further referred as "Aurora") is a corpus intended for the evaluation of front-end feature extraction algorithms in environments with various background noise conditions, It is also used more widely by researchers to evaluate and compare the performance of noise robust speech recognition algorithms [21].

The Aurora data is based on a version of *TIDigits* corpus downsampled to 8 kHz. Different noise signals have been digitally added to the clean speech data. The TIDigits corpus consists of data which was originally designed and collected at Texas Instruments, Inc. for the purpose of designing and evaluating algorithms for speaker-independent recognition of connected digit sequences. There are 326 speakers (111 men, 114 women, 50 boys and 51 girls), each pronouncing 77 digit sequences. Each speaker group is partitioned into disjoint training and test subsets [7].

The data used in the experiments of this thesis varies by noise type and signal-to-noise ratio (SNR). The following SNR conditions and noise environments were used in the experiments:

- Training subset: Clean, 20 dB, 15 dB, 10 dB, 5 dB
- Test subset: Clean, 20 db, 15 dB, 10 dB, 5 dB, 0 dB, -5 dB
- Noise environments: subway, babble, car noise, exhibition hall.

The training subset duration is approximately 15 minutes per condition (4 hours overall). The test subset duration is approximately 30 minutes per condition (14 hours overall).

We are not aware of publicly available ground truth labels for Aurora corpus. A typical approach of generating these labels is to annotate the "clean" subset by a VAD that is not involved in the study [32, 9].

We used the entropy-based VAD to generate the ground truth labels. The entropy-based VAD was chosen over energy-based VAD because of better performance in classifying dull speech sounds, such as "[ks]" and "[t]" endings in the "six" and "eight" words.

Visual and audible inspection was used to tune the entropy VAD threshold parameters in order to achieve good performance on given data. The VAD decision step is 10 ms. Thus, the resolution of the ground truth is also

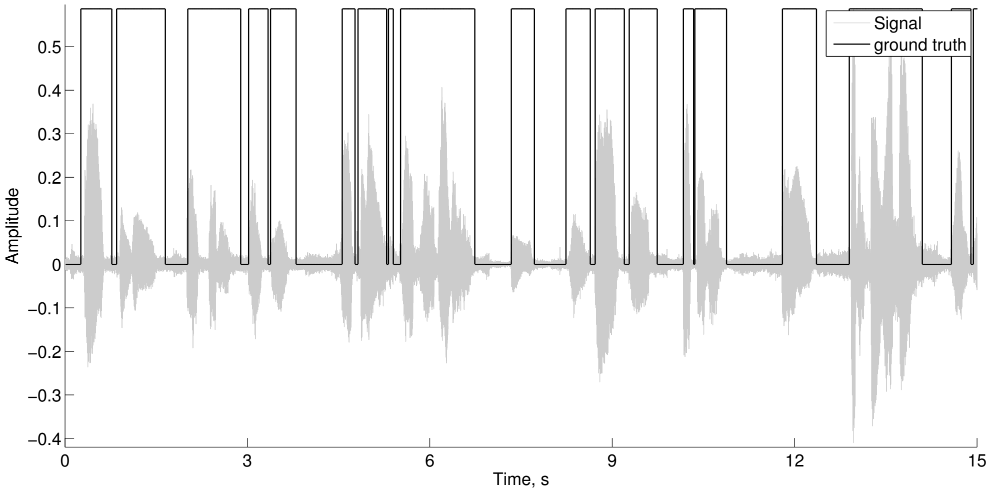


Figure 4.1: 15 seconds long excerpt from Aurora corpus. The signal contains subway noise added at SNR = 15 dB. The bars determine speech and non-speech segments of the signal, as defined in ground truth. Apparently not only the ground truth is meaningful, it precisely indicates the very short pauses between words.

one label per 10 ms. The MATLAB implementation of the entropy VAD is available in Appendix A.

One of the most important training data properties that could be calculated based on the ground truth labels is the *speech to non-speech ratio* (snsr) of the data:

$$\text{snsr} = \frac{\#\text{ of speech frames according to the GT}}{\#\text{ of nonspeech frames according to the GT}} \tag{4.1}$$

According to ground truth, the speech to non-speech ratio of Aurora corpus is 64% : 36%. An example of Aurora's signal waveform and corresponding GT is shown in Fig. 4.1.

4.1.2 NIST '05

The *NIST* (National Institute of Standards and Technology) *speaker recognition evaluation* (SRE) campaigns are affiliates of yearly evaluations conducted by NIST. The results of these evaluations help to find the right direction in which speech processing algorithms should be developed. [27]. In this thesis, we utilize speech data from NIST 2005 SRE corpus (referred to as "NIST '05").

The data was provided by LDC as the part of Mixer project. This project invited participating speakers to take part in numerous six-minute conversations on specified topics with strangers. Speakers were encouraged to use different telephone instruments for their initiated calls [27]. The audition of the data disclosed that the spoken speech is easy recognizable and the background noise is not high. Although there are various training and testing conditions in NIST'05 corpus, they are not specially targeted for VAD evaluation purposes.

The original two-channel data (one speaker per channel) was splitted and downsampled to 8kHz. The duration of training subset is approximately 4 hours. The duration of test subset is approximately 12 hours.

The VAD ground truth was extracted from automatic speech recognition (ASR) labels provided by NIST in the original corpus. The resolution of the ground truth is one label per 10 ms. According to the ground truth, the speech to non-speech ratio is 49% : 51%. This ratio is explained by the nature of the corpus: two men are speaking by the phone and the channels are recorded separately. Usually one of the speakers is silent listening to the other one speaking. This assumption leads to a conclusion that a half-part of both channels contains speech whereas another half contains background

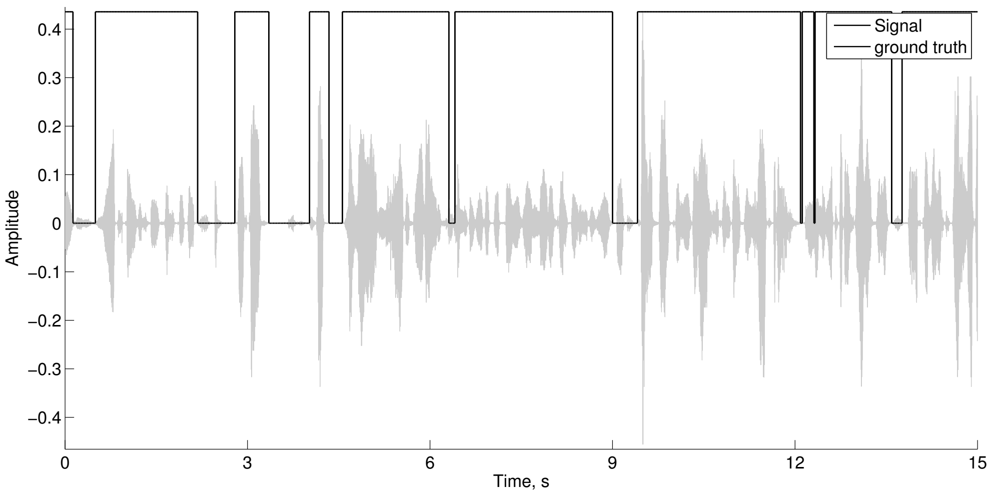


Figure 4.2: 15 seconds long excerpt from NIST'05 corpus. It is easy to notice that the magnitude of the background noise of this excerpt is smaller than the magnitude of the background noise of the signal shown in fig. 4.1.

noise only.

An example of NIST's signal waveform and corresponding ground truth is shown in Fig. 4.2.

4.1.3 Bus stop

The *Bus stop* corpus consists of short human speech commands and synthesized speech that provides rather long explanations about bus schedules (both in Finnish language) [43]. The audition of the data disclosed that the background noise is high, yet the spoken speech is recognizable.

The data was recorded as 8 kHz sampling rate. The duration of the training subset is approximately 45 minutes. The duration of the test subset is approximately 2 hours. The ground truth is human-labeled and the resolution

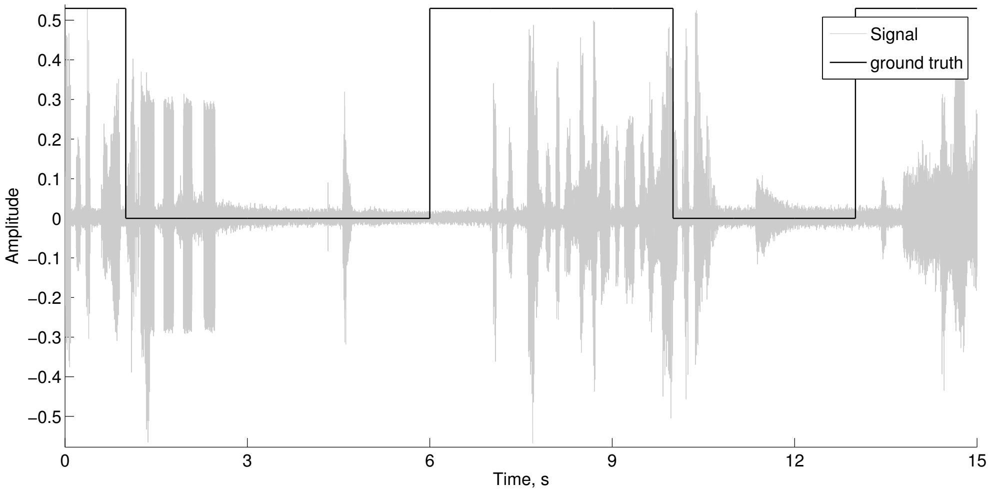


Figure 4.3: 15 seconds long excerpt from Bus stop corpus

is 1 label per second. According to the ground truth, the speech to non-speech ratio is about 80% : 20%. An example of Bus stop signal waveform and corresponding GT is shown in Fig. 4.3.

4.1.4 Lab

The *Lab* data set consists of a long continuous recording from the lounge of Speech and image processing unit (SIPU) at University of Eastern Finland (UEF). The goal of this corpus is to simulate wiretapping materials that are relevant in forensics.

The recording device is a Labtec PC microphone attached to a wall at a height of 1.8 m. The distance between the microphone and the speakers is about 4-6 meters. The audition of the data disclosed that the spoken speech is very low and hardly recognizable. The background noise is also low. From time to time, one car hear as a door opens, as a kettle boils, footsteps, keyboard

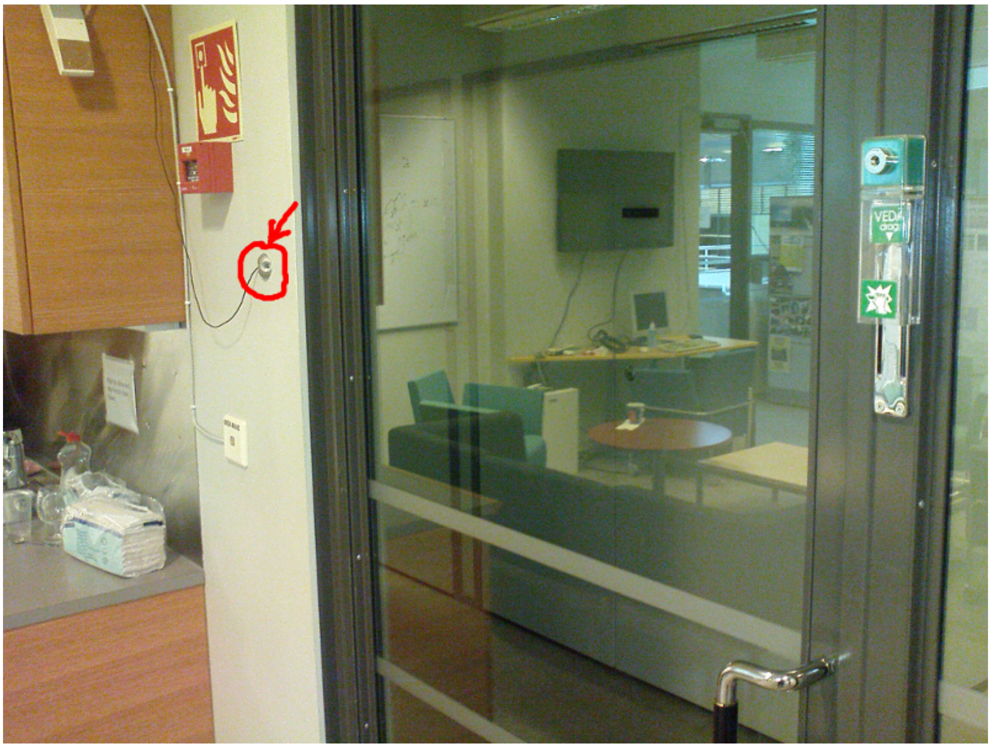


Figure 4.4: Labtec PC microphone at the entrance to the SIPU laboratory. Notice the reflection of the table at which the discussions are generally held.

clipping etc. The overall SNR is very low.

Prior to VAD analysis, the original 44.1 kHz data was downsampled to 8 kHz. The duration of training subset is 1 hours 45 minutes. The duration of test subset is 2 hours 45 minutes. Similar to the Bus stop data, the ground truth is human-labeled and the resolution is 1 label per second. According to the ground truth, the speech to non-speech ratio of the training data is 7% : 93%. The speech to non-speech ratio of the testing data is 13% : 87%. An example of Lab signal waveform and corresponding GT is shown in Fig. 4.5.

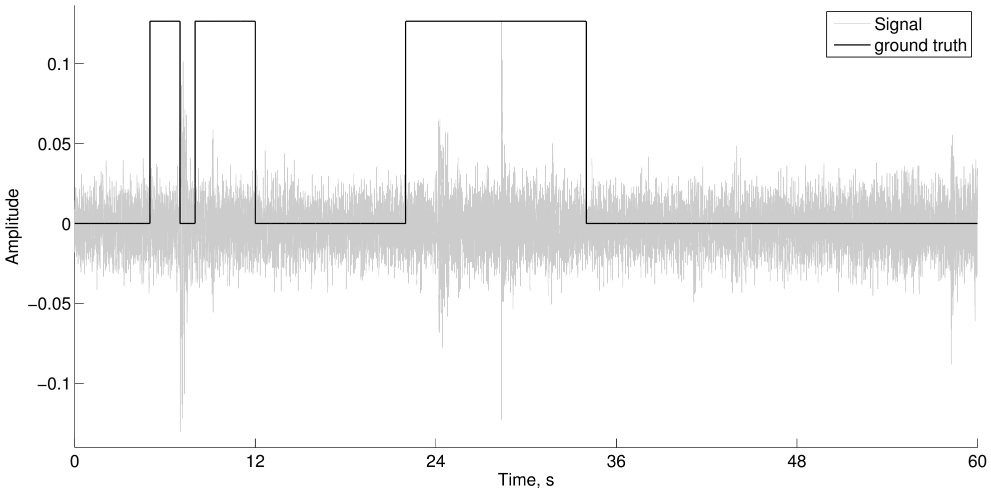


Figure 4.5: Lab corpus' 30 second long waveform example

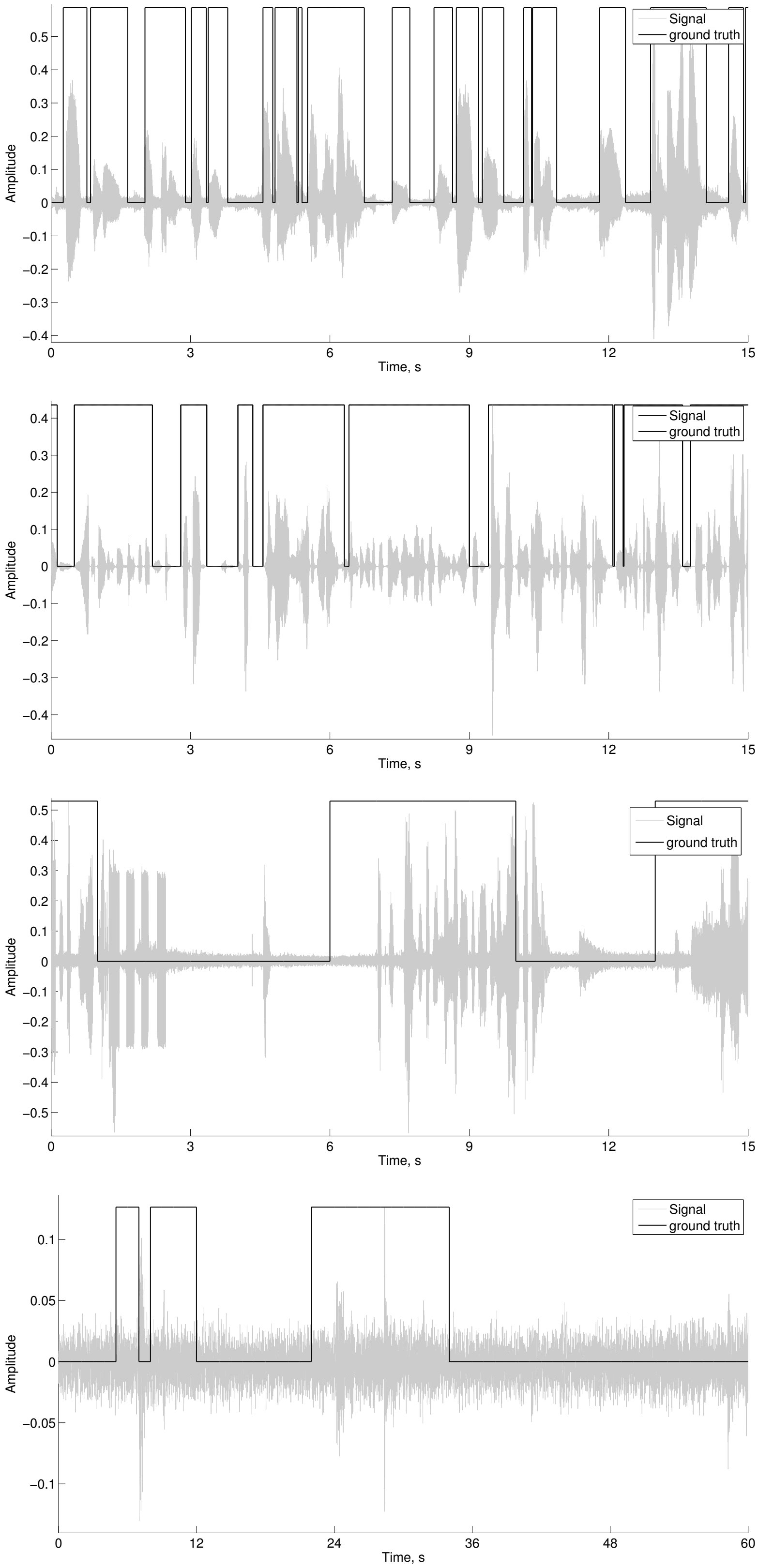


Figure 4.6: Evaluation corpora waveform and ground truth examples. Downright: Aurora, NIST, Bus stop, Labra

Table 4.1: Summary of the VAD evaluation corpora used in this thesis

Corpus	Aurora [21]	NIST'05 [27]	Bus stop [43]	Lab
Recording equipment	Electro-voice RE-16 dynamic cardiod microphone	Telephony conversations	Telephony speech commands	PC microphone
Environment	Studio + digitally simulated noises	Unknown	Outdoors	Indoors
Training data duration	4 hrs.	4 hrs.	45 min.	$1\frac{3}{4}$ hrs.
Test data duration	14 hrs.	12 hrs.	2 hrs.	$2\frac{3}{4}$ hrs.
Speech to non-speech ratio (training section)	64% : 36 %	49% : 51 %	80% : 20 %	7% : 93 %
Speech to non-speech ratio (test section)	61% : 39 %	48% : 52 %	78% : 22 %	13% : 87 %
Ground truth resolution	10 ms	10 ms	1 s	1 s

4.2 VAD algorithms

The following VADs are used in the experiments:

- Energy [Appendix A] - as an example of a very simple VAD algorithm.
- G.729 [3] - a well-known, but outdated algorithm.
- AMR1 and AMR2 [36] - modern algorithms widely used in audio compression
- SILK [39] - a widespread algorithm used in popular Skype program
- AFE [37] - an algorithm designed for special (distributed speech recognition) purpose.

The Entropy VAD (Appendix A) was used to annotate Aurora corpus (“clean” SNR cases), thus not used in further experiments.

4.3 Error metrics

4.3.1 Miss and False alarm rates

Measuring the performance of a VAD is a real challenge. Consider a VAD as a black box, which outputs binary labels, indicating speech and non-speech segments of an input signal. The ground truth of a corpus also consists of zeros and ones. To measure VAD performance we compare its output with the ground truth. A common way of presenting the predicted and actual classifications is *confusion matrix* [18].

The relations between confusion matrix values are:

Table 4.2: Confusion matrix

		Actual labels (ground truth)	
		Speech (1)	Non-speech (0)
VAD output (predicted labels)	Speech (1)	True Positive	False Positive
	Non-speech (0)	False Negative	True Negative

The confusion matrix does not differ much from the VAD relationship table shown earlier (Table. 3.1). A ground truth could be considered as the “absolute” VAD and the larger the correlation between a VAD and ground truth is, the better.

- The sum of all confusion matrix values corresponds to the total number of labeled frames. It is assumed, that the amount of ground truth and VAD frames is equal.
- # of true positives + # of false negatives = # of frames labeled as speech in ground truth
- # of false positives + # of true negatives = # of frames labeled as non-speech in ground truth
- # of true positives + # of false positives = # of frames labeled as speech by the VAD
- # of false negatives + # of true negatives = # of frames labeled as non-speech by the VAD

The *miss rate* (MR) shows the amount of speech data missed by a VAD. A low MR is crucial for applications that require the “full picture” of speech data, with possibly many non-speech segments included, like forensics. MR is defined as follows:

$$\text{MR} = \frac{\text{False negatives}}{\text{False negatives} + \text{True positives}} \tag{4.2}$$

The *false alarm rate* (FAR), shows the proprotion of non-speech data misclassified as speech. FAR is defined as follows:

$$\text{FAR} = \frac{\text{False positives}}{\text{False positives} + \text{True negaives}} \tag{4.3}$$

An example of MR and FAR plots is shown in Fig. 4.7.

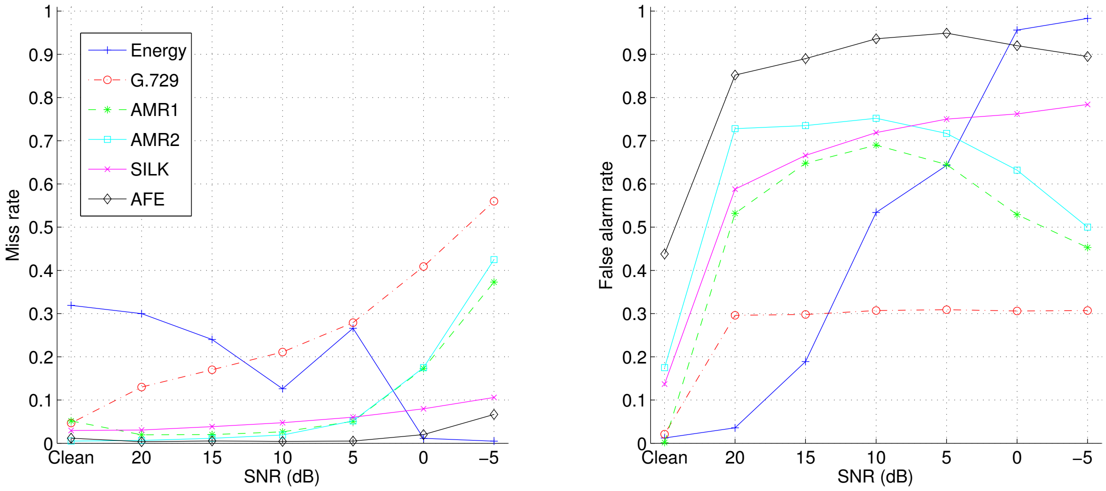


Figure 4.7: Miss rate and False alarm rate plots. Although G729's false alarm rate is almost independent of SNR, its miss rate is much higher than AMR's.

4.3.2 Total error rate

Both miss rate and false alarm rate provide enough data to describe a VAD performance. The total error rate (TER) shows the total proportion of wrong

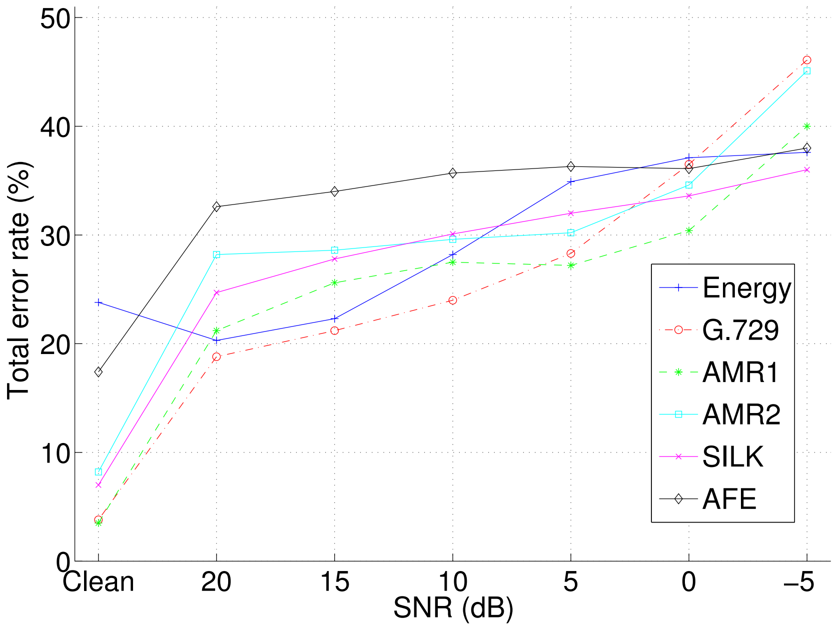


Figure 4.8: Total error rate plot

decisions. TER is defined as follows:

$$\text{TER} = \frac{\text{Number of false decisions}}{\text{Total number of frames}} \tag{4.4}$$

Although TER provides a single metric to compare VAD algorithms performance, it hides the full picture of how VAD behaves in various conditions. Both miss rate and false alarm rate should be taken into consideration before making a performance decision based on total error rate.

4.4 VAD$_{py}$

The experiments carried out in this thesis required a framework which would allow to use the analyzed corpora and different VAD algorithms from one side and various analysis metrics from the other. Since there was no required software available, a new framework was designed and written from scratch. It is called VAD$_{py}$.

VAD$_{py}$ is a universal, modular, easy-to-run and easy-to-extend voice activity detection algorithms evaluation framework that integrates different corpora, VADs, error metrics and performance reports in one platform. VAD$_{py}$ is written in Python [1], which was chosen for its high-level programming language capabilites and because it is loved by the author of this thesis. The source code is available at `http://code.google.com/p/vadpy/`.

The basic concepts of VAD$_{py}$ are the *pipeline*, the elements and the modules. These concepts were borrowed from the GStreamer project [2]. One can think of a pipeline, as of a factory pipeline. A pipeline has a *source* at its head, which "drops" the *elements* on the pipeline. Initially an element has the basic information on it, e.g. corpus name, the paths to the data and ground truth files and data description (e.g. data sampling rate, bit rate etc.). The *modules* modify the elements one-by-one by processing the information that is already attached to the element and adding the processing results back to the element. For example, a module reads the data path attached to the element, runs a VAD algorithm on the data and attaches the VAD output file's path to the element.

A typical command-line execution of VAD$_{py}$ looks like following:

[1] `http://python.org`

[2] `http://www.gstreamer.net/`

```
vad.py ! aurora snr=20 ! iaurora ! g729 ! ig729 ! confusion
>> Miss rate (%):               13.0
>> False alarm rate (%):        29.6
```

- **aurora** module is the source of the data in the pipeline. It represents Aurora corpus, and fills the pipeline with Aurora's data files and GT files' paths (one data file path and corresponding GT file path per element). **snr** is an option which tells the module to use the "SNR=20dB" data only.

- **iaurora** is Aurora's ground truth files parser. The output of this module are the corpus-independent labels internally used in VAD$_{py}$. The labels are attached to corresponding elements.

- **g729** is the module that executes G.729 VAD over each element's data. The output path of the VAD's result is attached to the element.

- **ig729** is the G.729's output parser. It parses the VAD output located by path attached in previous step. The parsed labels are attached to the corresponding elements.

- Finally, **confusion** compares the GT and VAD output by means of confusion matrix. The module summarizes the results from multiple elements, computes the mean errors and writes a formatted output to stdout.

Chapter 5

Experiments

5.1 Individual VAD performance

Performance of the individual VADs forms a baseline for any further improvements by fusion techniques. Figure 5.1 shows the miss and false alarm rates for all the considered VADs on the Aurora 2 corpus. Figure 5.2 shows the corresponding total error rate (TER) plot and the TER values averaged over different Aurora corpus conditions for a given signal-to-noise ratio (SNR).

We make the following observations:

- The energy VAD has satisfactory performance in high SNR conditions, with a high miss rate (approx. 30%). The performance of the energy VAD drops consequently with decreasing SNR, where nearly every frame is classified as non-speech.
- The G.729 VAD has a very stable false alarm rate, outperforming all the other VADs for SNRs below 15dB. But the miss rate remains high, up

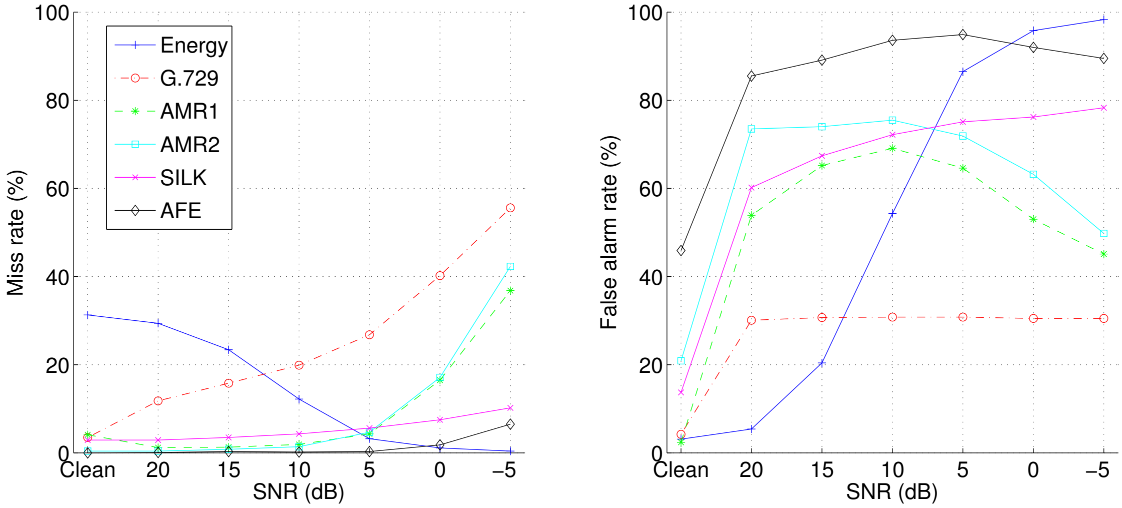
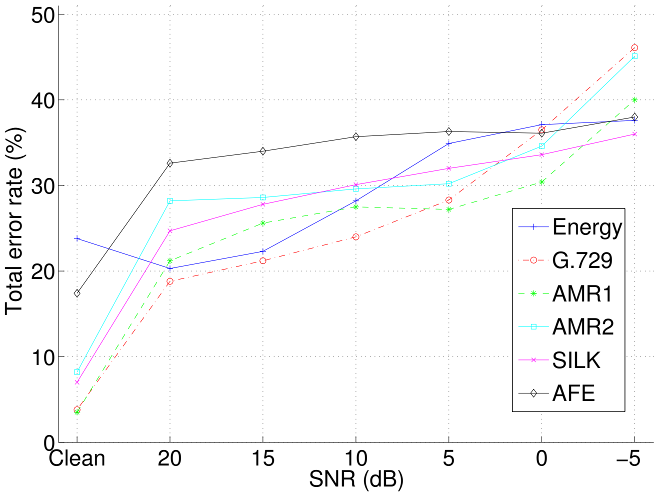


Figure 5.1: Miss rate and false alarm rates plots for VADs evaluation on Aurora corpus. The VAD with the best average performance has the smallest area under the corresponding total error rate graph

to 55%, in low signal-to-noise conditions below 0 dB. Although G.729 holds the second rank according to the average total error rate value (Fig. 5.2), it should not be used in applications that aim for a low miss rate.

- The AMR1 and AMR2 VADs have about the same behaviour in classifying speech correctly unless the SNR drops to -5 dB level. From the false alarm graph, it is clear that AMR1 outperforms AMR2 by 5% to 20% under different SNRs. AMR1 is the best VAD according to the average total error rate value.

- The SILK VAD seems to outperform most of the VADs, since both miss and false alarm rates' curves increase linearly and do not have sudden steep “hills” or “valleys” as AMR1 and AMR2 do.



VAD	TER (%)
Energy	29.1
G.729	25.5
AMR1	**25.0**
AMR2	29.2
SILK	27.3
AFE	32.87

Figure 5.2: Total error rate plot and average TER over SNR conditions values on Aurora corpus

- The performance of AFE VAD is very low on Aurora, yielding the lowest performance among all VADs. Because AFE VAD has a quaternary VAD output, the problem of selecting a static or adaptive threshold to convert this output to binary "speech" and "non-speech" labels is a separate problem that was not studied deeply in this thesis. In this thesis, we mapped the 4-level output of AFE as was explained in Section 2.4.3. Due to its low performance, AFE will be excluded from further experiments.

5.2 Majority voting in VAD combinations

The simple majority voting scheme requires an odd number of VAD "votes". Thus, it is possible to make ten three-VAD combinations out of 5 available VADs.

Table 5.1: ρ-correlation based on Aurora training set

	G.729	AMR1	AMR2	SILK
Energy	0.313	0.287	0.216	0.356
G.729		0.353	0.250	0.387
AMR1			0.716	0.683
AMR2				0.630

The large ρ value is, the larger is the correlation between VADs.

As discussed in Section 3.1, it is desirable to combine the least correlated VADs. The correlation coefficients (ρ) between our VADs (Eq. 3.1) are shown in Table 5.1. The large the value of ρ for a given VAD pair is, the larger is the correlation. We define overall heuristic correlation measure by summing up the VAD ρ correlations. For example, the correlation coefficient between energy, AMR1 and G.729 is computed as following:

$$\rho_{\text{energy,AMR1,G.729}} = \frac{1}{3}(\rho_{\text{energy,AMR1}} + \rho_{\text{energy,G.729}} + \rho_{\text{G.729,AMR1}}) \qquad (5.1)$$

According to Table 5.1, the least correlated VAD triplet consists of energy, G.729 and AMR2 VADs, the second best combination has AMR1 instead of AMR2 and the most correlated VAD triplet is AMR1, AMR2 and SILK. Theoretically, the larger the ρ-correlation is, the less performance boost could be achieved by combining the VADs, as the results produced by them are similar. Low correlation value of the energy, G.729 and AMR VADs is expected as the VADs utilize different algorithms and hangover schemes and energy VAD has no hangover scheme at all. Figures 5.3 and 5.4 show the results of majority voting carried out using several VAD teams.

The results of the majority voting experiments indicate that VAD combi-

Table 5.2: VAD triplets sorted by the smallest average pairwise ρ-correlation

Energy, G.729, AMR2	0.26
Energy, G.729, AMR1*	0.31
Energy, G.729, SILK	0.35
Energy, AMR2, SILK*	0.4
G.729, AMR2, SILK*	0.41
G.729, AMR1, AMR2	0.43
Energy, AMR1, AMR2	0.44
Energy, AMR1, SILK	0.44
G.729, AMR1, SILK	0.47
AMR1, AMR2, SILK*	0.67

The MR, FAR and TER of the teams marked by * are shown in Fig. 5.3 and Fig. 5.4.

nations outperform standalone VADs under certain SNR conditions. Four combinations were chosen as a subject to a detailed analysis:

- Energy, G.729, AMR1
- G.729, AMR2, SILK
- Energy, AMR2, SILK
- AMR1, AMR2, SILK

The energy, G.729 and AMR1 combination has the highest accuracy among the other combinations in high SNR condition but it fails in low SNR environment. This behaviour can be explained by the poor performance of the

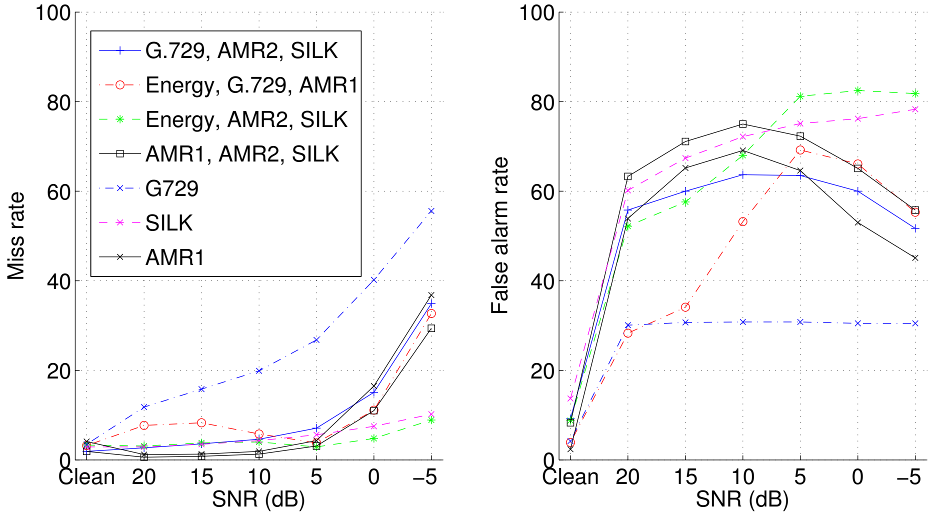


Figure 5.3: Miss rate and false alarm rate graphs of majority voting evaluation on Aurora corpus.

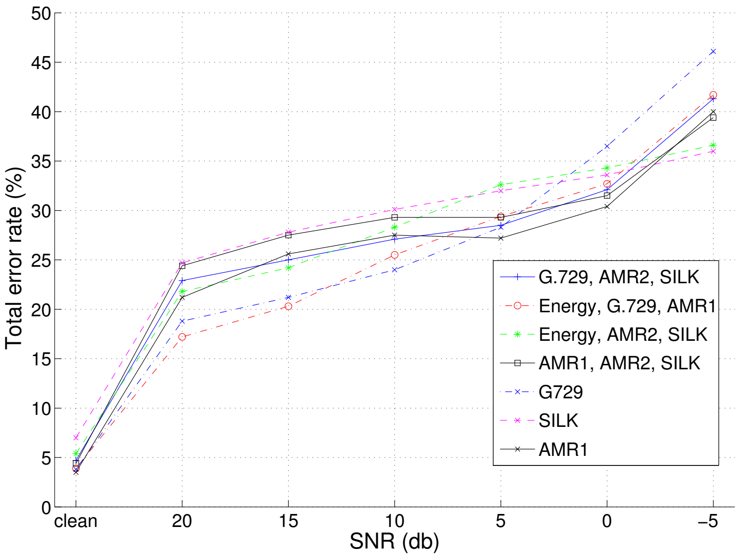


Figure 5.4: Total error rate of majority voting evaluation on Aurora corpus

energy VAD in low SNR environment. The analysis of this triplet's results is a good example when total error rate and miss/false alarm rates results should be analyzed together: The MR and FAR graphs (Fig. 5.3) of this combination have very sharp ascents and descends as the SNR conditions change which are not visible on the total error rate graph (Fig. 5.4).

According to the total error rate plot (Fig. 5.4), G.729, AMR2 and SILK is the next best triplet among the analyzed combinations. The analysis of the triplet's MR and FAR results shows, that the combination has stable behaviour in 20-5dB SNR conditions: the error rates are increasing in a linear fashion.

Energy, AMR2 and SILK is the second triplet with energy VAD in it. Its false alarm rate graph has the shape as energy, G.729 and AMR1 combination's FAR graph. This leads to an idea that it is the energy VAD which is responsible for the steep non-linear false alarm error rates growth along with the SNR. SILK plays a role in keeping miss rate low even in high SNR conditions.

The AMR1, AMR2, SILK team has the lowest performance in comparison with the other VAD triplets in SNR $\in (0-10]$ dB range, which agrees with the predictions based on ρ-correlation coefficient (Table 5.2). On the other hand, this combination outperforms the others analyzed combinations in SNR=0dB, which does not conform with the correlation-based performance assumption.

The results of the majority voting experiments indicate that it is possible to find a VAD combination that will outperform standalone VADs in certain SNR conditions.

Another reason to combine different VADs and to use the majority voting

Figure 5.5: Average TER of VAD combinations majority voting evaluation on Aurora corpus

VAD	**TER (%)**
G.729, AMR2, SILK	25.9
Energy, G.729, AMR1	**24.3**
Energy, AMR2, SILK	26.1
AMR1, AMR2, SILK	26.5
G729	25.5
SILK	27.3
AMR1	25.0

Although energy, G.729 and AMR1 combination outperforms other combination only in two SNR conditions (Table 5.3), it has the smallest average total error rate. This combination is the best choice if the SNR conditions are unknown and the best generic solution is required.

scheme is to obtain such combination that would have the best average performance in the conditions of interest.

Table 5.3: The results of majority voting experiments (total error rates (%))

SNR conditions VAD combinations	Clean	20dB	15dB	10dB	5dB	0dB	-5dB	Average
G.729, AMR2, SILK	4.7	22.9	25.0	27.1	28.5	32.1	41.3	25.9
Energy, G.729, AMR1	3.9	**17.2**	**20.3**	25.5	29.4	32.7	41.7	**24.3**
Energy, AMR2, SILK	5.4	21.8	24.2	28.3	32.6	34.3	36.6	26.1
AMR1, AMR2, SILK	4.4	24.4	27.5	29.3	29.3	31.5	39.4	26.5
G.729	3.8	18.8	21.2	**24.0**	28.3	36.5	46.1	25.5
SILK	7.0	24.7	27.8	30.1	32.0	33.6	**36.0**	27.3
AMR1	**3.5**	21.2	25.6	27.5	**27.2**	**30.4**	40.0	25.0
Average	4.7	21.5	24.5	27.4	29.6	33.0	40.1	25.8

Although, the analyzed VAD combinations do not outperform the standalone VADs in all SNR conditions, the experiments show that it is possible to decrease the total error rate of a VAD by combining it's results with other VADs. The lowest total error rate values for certain SNR conditions are marked via bold font.

5.3 Majority voting in standalone VAD's temporal context

Before proceeding to experiments with VAD combinations and majority voting in temporal context, we would like to study the effects of using the temporal context information of a standalone VAD. In the experiments, the temporal context size (Section 3.5) varies as $d \in \{0, 1, 2, 3, 5, 10\}$ and is measured in frames. The $d = 0$ graphs represent the error rates of the base individual VADs evaluation, which simplifies the comparison of the experiment's results.

Figure 5.6 shows the result of energy VAD evaluation over Aurora corpus. The effect of temporal context majority voting is observed on both miss and false alarm rate plots. Generally the miss rate decreases and the false alarm rate increases as the temporal context size increases. This behaviour is expected since the energy VAD misses natural sounds that are similar to noise (e.g. hissing sounds). Thus, the majority voting scheme in a temporal context should level off these gaps and decrease final decision's miss rates. On the other hand, the scheme increases the false alarm rate due to larger number of false decisions before the beginning and end of every word. In the clean SNR conditions, the total error rate is decreased from 23.8% to 20.6% when $d = 1$ and drops to 19.4% for $d = 10$. The performance improvement also appears in lower SNR conditions, but does not exceed 1% (Table 5.4). Finally, the temporal context majority voting approach can be considered as a basic hangover mechanism for the energy VAD due to its absence in the original implementation (Appendix A).

Figure 5.7 shows the result of AMR1 VAD evaluation over Aurora corpus.

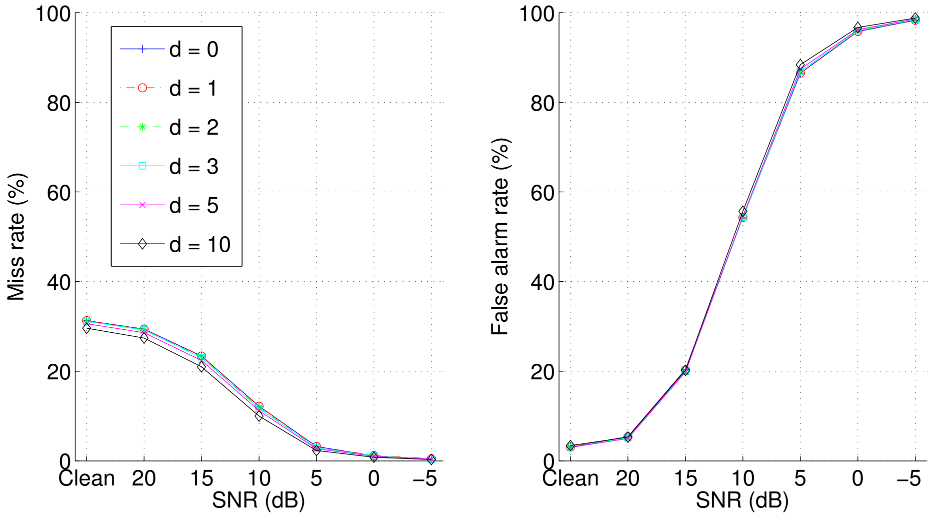


Figure 5.6: Miss rate and false alarm error rates graphs of energy VAD temporal context evaluation on Aurora corpus

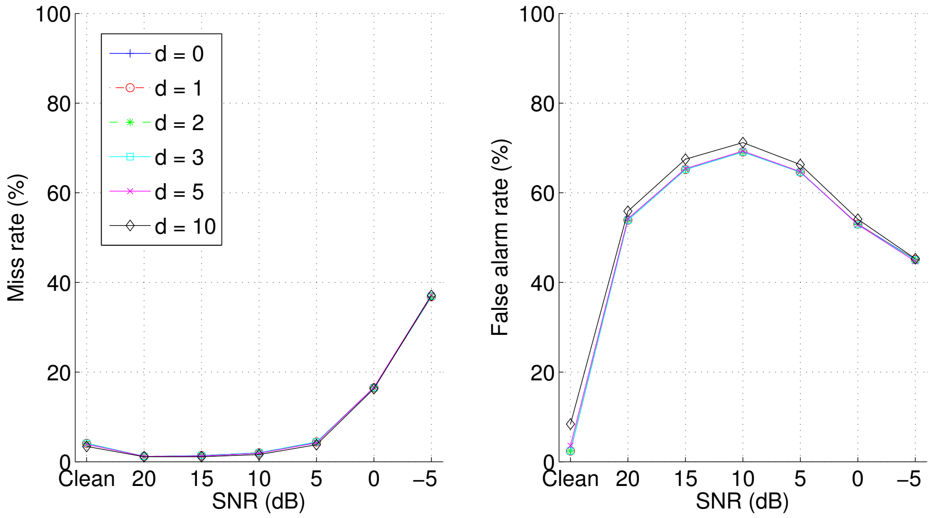


Figure 5.7: Miss rate, false alarm and total error rates graphs of AMR1 VAD temporal context evaluation on Aurora corpus

The expected behaviour of temporal context majority voting is negative due to previously used hangover scheme in AMR VAD algorithm (Section 2.4.2). The effect is mainly observed on false alarms, which increase with longer temporal context windows. In clean SNR condtions, the total error rate is decreased from 3.5% to 3.4% when $d = 2$ and climbs up to 5.3% for $d = 10$. The behavior of small performance improvement in small temporal context conditions and its declining in larger temporal context conditions is also observed in lower SNR conditions (Table 5.4).

The method has a small, yet negative impact on the base VADs performance results. The stable miss rate could be explained as the result of the internal hangover scheme, which is especially used to detect low power endings of speech bursts [36]. The overall effect of temporal context majority voting is shown in Table 5.4.

Table 5.4: The results of standalone VAD temporal context majority voting experiments (total error rate (%))

VAD	Clean signal					
Context size	**0**	**1**	**2**	**3**	**5**	**10**
Energy	23.8	20.6	20.5	20.4	20.1	**19.6**
G.729	3.8	3.8	3.6	**3.5**	3.9	5.3
AMR1	3.5	3.5	**3.4**	3.4	3.8	5.3
AMR2	**8.2**	8.2	8.2	8.2	8.3	8.7
SILK	**7.0**	7.0	7.0	7.0	7.1	8.1
Added noise at 15dB						
Energy	22.3	22.2	22.1	21.9	21.4	**20.7**
G.729	21.2	21.2	21.1	21.0	**20.7**	31.5
AMR1	**25.6**	25.6	25.6	25.6	25.6	26.3
AMR2	**28.6**	28.6	28.6	28.6	28.7	29.2
SILK	**27.8**	27.8	27.8	27.8	27.8	28.4
Added noise at 5dB						
Energy	33.6	33.6	33.6	33.6	**33.5**	33.6
G.729	36.5	36.5	36.4	36.3	**36.2**	36.7
AMR1	30.4	30.4	**30.3**	30.3	30.4	30.6
AMR2	**34.6**	34.6	34.6	34.6	34.7	35.2
SILK	**33.6**	33.6	33.6	33.6	33.5	33.6

According to the total error rate results, the performance of industrial VADs improves in $d \in [1,3]$ range. The performance of the energy VAD as well improves in $d \in [1,10]$. The lowest total error rate values for the smallest possible temporal context size are marked via bold font.

5.4 Majority voting in VAD combination's temporal context

The following experiments are based on the results of previous majority voting (Section 5.2) and temporal context voting experiments (Section 5.3).

We expect that the usage of temporal information will increase the performance of combinations that contain energy VAD and will not affect or decrease the performance of triplets that consist of VADs with built-in hangover mechanism. The analyzed VAD combinations are energy, G.729, AMR1 and G.729, AMR2, SILK.

The results of the experiments are shown in Table 5.5. The performance of energy, G.729 and AMR1 is increased by 1.4%. The performance of G.729, AMR2 and SILK combination remains in ±0.2% range of the base result values shown without utilizing the temporal context.

Table 5.5: The results of standalone VAD temporal context majority voting experiment's (average total error rate (%))

Context size **Combination**	0	1	2	3	5
Energy, G.729, AMR1	24.3	23.0	22.9	**22.8**	22.8
G.729, AMR2, SILK	25.9	25.9	**25.8**	25.9	26.0
AMR1	**25.0**	25.0	25.0	25.2	25.4

5.5 Histogram model-based fusion

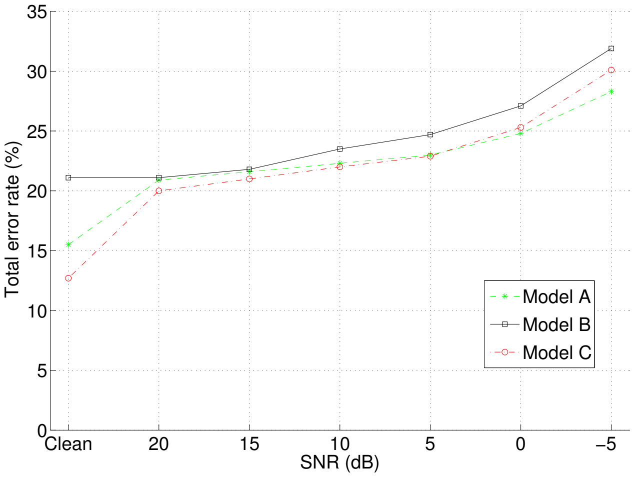
The histogram model-based fusion is a method which utilizes the prior knowledge and analysis of the data. None of the methods used in previous experiments had this knowledge, thus an overall performance increase of the experiments' results is expected. The VAD combinations analyzed in this section are the same as in Section 5.2:

- Energy, G.729, AMR1
- G.729, AMR2, SILK
- Energy, AMR2, SILK
- AMR1, AMR2, SILK

As mentioned in Section 4.1.1, the training setup for Aurora corpus consists of the following SNR conditions: Clean, 20dB, 15dB, 10dB and 5dB. The speech and non-speech histograms that form the basis of the speech activity model could be built from the whole set or a subset of the training data. To find out whether the full set of the training data or a smaller subset could be used to obtain the best performance, an experiment with three different speech activity models was carried out.

The models used in the experiment are:

- **Model A** - is a generic model based on all available training data.
- **Model B** - is based on the data from the mid range of available training SNR conditions: 20db, 15db, 10db.



VAD	TER (%)
Model A	22.3
Model B	24.4
Model C	**22.0**

Figure 5.8: Energy, AMR2 and SILK triplet evaluation with various speech activity models and the mean values of TER over SNR conditions

Model A is based on all available training data; Model B is based on the data with SNR ∈ {20dB, 15dB, 10dB}; Model C is based on the data with SNR ∈ {Clean, 15dB, 5dB}.

- **Model C** is based on the extreme Clean, 5dB and middle 15dB SNR conditions.

Figure 5.8 shows the results of evaluating energy, AMR2 and SILK combination with the A, B and C models.

Model B shows the worst performance in the experiment. Most notably it fails in clean SNR environment, as no preliminary knowledge of clean speech was given. The model also fails to outperform models A and C in 20dB - 10dB SNR conditions, which is the fault of a mismatch between test data's and model's training histograms.

Model A is based on the idea that "the more prior knowledge we have - the better". It is meant to be generic and independent of SNR conditions.

Although being the second best in the [Clean, 10dB] range, it caches up and outperforms model C in [5dB, -5dB] SNR conditions.

Model C verifies if the preliminary knowledge of data from boundary and mean SNR conditions is enough to obtain the same performance as Model A. Model C outperforms model A by 0.3% of average total error rate result. This is a small number and the further model selection requirements should include the size of training data set (smaller is better) and the time spent in training phase (which is smaller for model C). We have chosen model C approach for our experiments.

The results of the experiments showed that histogram model-based approach dramatically improves VAD combinations' performance in comparison to standalone VADs and majority voting-based triplets' results.

Figures 5.9 and 5.10 show the results of histogram model-based fusion for the analyzed VAD triplets.

Comparing to the best results of standalone VADs and majority voting-based combinations' results, the histogram model-based method decreases the total error rate by 1-10% in various SNR conditions. The lowest average total error rate is obtained by energy, G.729 and AMR1 combination, which is 19.9%. This is 5% smaller than the best majority voting or standalone VAD average TER result.

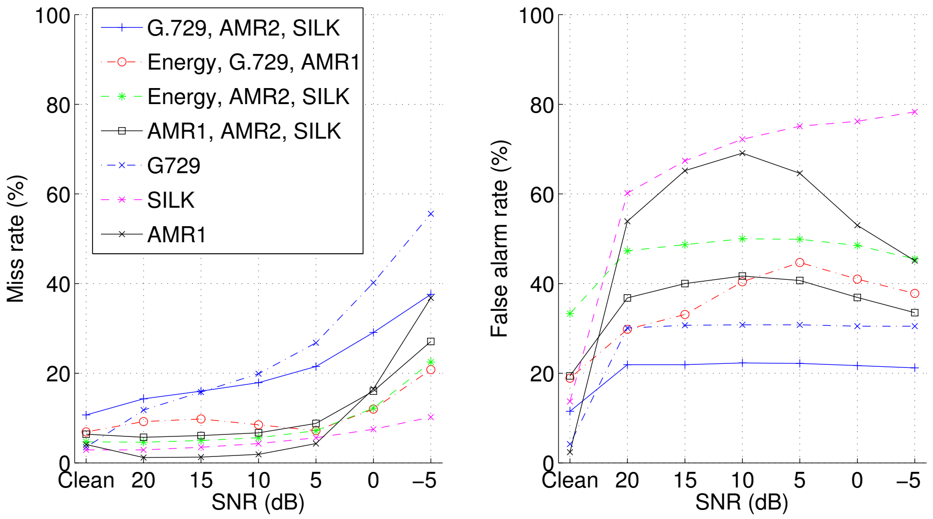


Figure 5.9: Miss rate and false alarm error rates graphs of VAD triplets histogram model-based evaluation on Aurora corpus

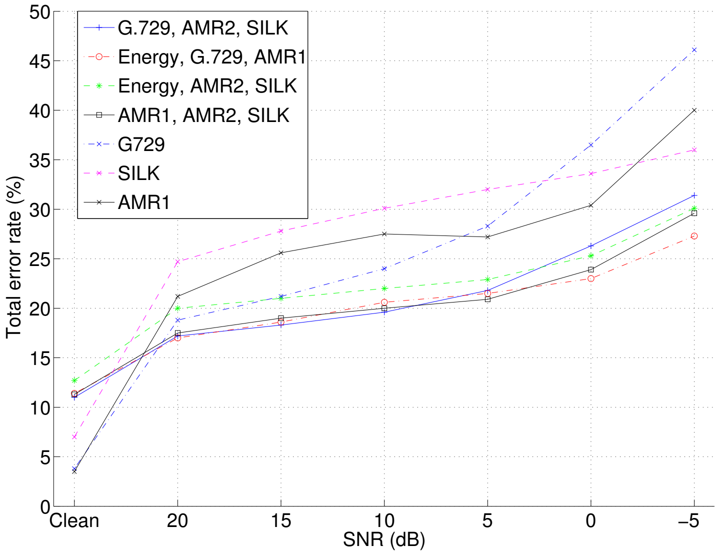


Figure 5.10: Total error rate graph of VAD triplets histogram model-based evaluation on Aurora corpus

Table 5.6: The results of histogram model-based fusion voting experiments (total error rate (%))

SNR conditions VAD combinations	Clean	20dB	15dB	10dB	5dB	0dB	-5dB	Average
G.729, AMR2, SILK	11.0	17.2	**18.3**	**19.6**	21.8	26.3	31.4	20.8
Energy, G.729, AMR1	11.4	**17.0**	18.6	20.6	21.5	**23.0**	**27.3**	**19.9**
Energy, AMR2, SILK	12.7	20.0	21.0	22.0	22.9	25.3	30.1	22.0
AMR1, AMR2, SILK	11.3	17.5	19.0	20.0	**20.9**	23.9	29.6	20.3
G.729	3.8	18.8	21.2	24.0	28.3	36.5	46.1	25.5
SILK	7.0	24.7	27.8	30.1	32.0	33.6	36.0	27.3
AMR1	**3.5**	21.2	25.6	27.5	27.2	30.4	40.0	25.0
Average	8.6	19.4	21.6	23.4	24.9	28.4	35.3	22.9

VAD combinations fused via histogram-based method outperform standalone VADs almost in all, but clean conditions. The lowest total error rate values for certain SNR conditions are marked via bold font.

5.6 Experiments on NIST, Busstop and Lab corpora

The goal of the experiments with the rest of the corpora is to verify the conclusions made during the experiments with Aurora corpus. The key difference between Aurora corpus and the rest of the corpora is that Aurora's data includes a range of SNR conditions, which are not available in NIST, Busstop and Lab. Thus, for comparison purposes the results of this section could be interpreted as average error rates from experiments on Aurora corpus.

5.6.1 Experiments on NIST corpus

The results of standalone and histogram model-based fusion experiments on NIST corpus are shown in Table 5.7. The best standalone VAD is AMR1 with 29.8% total error rate. All analyzed VAD combinations show smaller TER and AMR1, AMR2 and SILK ahead of the pack with the lowest 26.4% total error rate. These results confirm the performance increment by the histogram model-based fusion method.

5.6.2 Experiments on Bus stop corpus

Bus stop corpus is the first of the corpora with 1 second ground truth resolution. The speech-to-non-speech ratio of the corpus is 78 % : 22 %, it is predictable that VADs miss rate can be low due to prevailing “speech” labels and false alarms will not affect the total error rate in the same proportion as misses.

Table 5.7: The results the experiments on NIST corpus (total error rate (%))

Error rates VAD combinations	TER	MR	FAR
Energy	38.9	63.9	**14.9**
G.729	31.3	22.1	40.0
SILK	37.0	20.0	53.3
AMR1	**29.8**	25.0	34.4
AMR2	33.2	**19.1**	47.8
G.729, AMR2, SILK	27.7	23.2	32.1
Energy, G.729, AMR1	26.9	21.9	31.8
Energy, AMR2, SILK	29.3	**20.2**	38.0
AMR1, AMR2, SILK	**26.4**	22.7	**30.0**
G.729, AMR1, SILK	26.8	22.3	31.3
Energy, G729, AMR2	27.8	21.5	34.0

Table 5.8: The results of the experiments on Bus stop corpus (total error rate (%))

Error rates VAD combinations	TER	MR	FAR
Energy	31.4	32.9	25.3
G.729	26.3	22.4	41.5
SILK	27.2	20.9	51.5
AMR1	**19.4**	**16.1**	**32.4**
AMR2	19.8	14.0	42.3
G.729, AMR2, SILK	19.3	13.1	**43.1**
Energy, AMR2, SILK	18.7	12.0	44.2
Energy, G.729, AMR1	18.8	11.8	45.8
AMR1, AMR2, SILK	**17.7**	**10.8**	44.3

The results should be compared among themselves only, due to inaccurate ground truth.

Further analysis of the corpus' ground truth showed that the part of the data was not labeled accurately. The misclassified sections are 1 to 3 seconds long. The analysis frame size varies between 10 - 30 milliseconds for different VADs. This means that there are 30 - 100 inaccurate training frames for a VAD combination in the histogram model-based method. Although, the results of standalone or VAD combinations runs on the corpus should not be interpreted as correct ones, they still can be compared among themselves. The results of standalone and histogram model-based fusion experiments are shown in Table 5.8.

AMR1 is the best standalone VAD with 19.4% total error rate. Histogram model-based test results of all VAD combinations show smaller total error rate, that AMR1. AMR1, AMR2 and SILK combination shows the lowest 17.7% total error rate.

5.6.3 Experiments on Lab corpus

Lab is the second corpus with 1 second ground truth resolution. There is no expressed background noise in the data, yet the SNR of the signal is low, due to a weak sensitivity and of the recording microphone.

The analysis of Lab corpus' ground truth showed that its accuracy leaves much to be desired. The misclassified segments are 1-6 seconds long. Considering the small amount of speech data in training conditions (Table 4.1) and a prevailing number of misclassified segments in ground truth, it was unlikely that histogram model-based method would show any performance improvement comparing to standalone VADs.

The results of the experiments showed that the histogram model-based method cannot be used with the given low-quality ground truth and the “fallback” majority voting method did not give any further improvement to the performance of the standalone VADs (Table 5.9). Unfortunately the results of the experiments carried out on Lab corpus cannot be considered as reliable ones.

5.7 Discussion

During the research we discovered that every standalone VAD has an individual behaviour in different conditions and non of the analyzed VADs

Table 5.9: The results of the experiments on Lab corpus (total error rate (%))

Error rates VAD combinations	TER	MR	FAR
Energy	35.9	70.9	30.8
G.729	19.0	65.3	12.2
SILK	37.2	37.2	37.2
AMR1	15.5	63.8	8.5
AMR2	19.2	46.6	15.2
Energy, AMR2, SILK*	25.0	47.8	21.6
AMR1, AMR2, SILK*	18.3	51.4	13.5
AMR1, AMR2, SILK**	12.7	100.0	0.0

(*) - majority voting method; (**) - histogram model-based method. The combination used in histogram model-based method completely fails in detecting speech. The combinations used in majority voting method cannot outperform the best standalone VAD (AMR1) results. The results should be compared among themselves only, due to inaccurate ground truth.

outperforms the others in all conditions. The VAD with the best average performance the is AMR1.

The temporal context majority voting method is applicable as a very basic hangover mechanism for VAD algorithms that lack one. It should not be used with VADs that already utilize temporal information to avoid performance loss.

The majority voting method gives a small performance boost for a limited number of VAD combinations. The target applications of this method are those, where a new VAD behaviour in terms of miss rate and false alarm rate is desired. The majority voting method in VAD combination's temporal context improves the performance of combinations with energy VAD in them by 1%-2%. The reason is the improvement of the energy VAD's result due to the absence of a hangover mechanism in the original algorithm's implementation.

The histogram model-based method improves the standalone VAD results by 2%-5%. Unlike the other fusion experiments, all VAD combinations outperform AMR1 when used via histogram model-based method. Although AMR1, AMR2 and SILK shows the best results on NIST and Bus stop corpora, energy, G.729 and AMR1 triplet is still considered superior due to the best results shown on Aurora 2 corpus.

Chapter 6

Conclusion

The aim of this work was to study standalone VADs behaviour and analyze fusion methods that could combine VADs' outputs in order to achieve better speech classification performance. Eight VADs, energy-based, entropy-based, G.729, AMR1, AMR2, SILK and AFE were used during the research, five of them, were analyzed in-detail and used for fusion experiments. To accelerate the research process a complex VAD algorithms testing framework was written from scratch. During the research three fusion methods were applied and the results were presented discussed in detail. Four data corpora were used in experiments. The well-known Aurora 2 corpus was used as the base for research experiments and NIST'05, Bus stop and Lab corpora were used to confirm the achievements of made on Aurora corpus.

The standalone AMR1 VAD showed the smallest average error rate in all experiments and is considered the best VAD among energy-based, G.729, AMR2 and SILK. The fusion methods that involve temporal context improved the results of energy VAD and combinations which included energy VAD. The majority voting in temporal context could be used as a simple hangover mechanism for energy VAD.

In order to predict the best VAD combinations for majority voting fusion the VAD correlation test was successfully applied. In majority voting experiments the analyzed VAD combinations showed small performance increase, yet the method could be effectively used in order to achieve a new VAD behaviour in various signal-to-noise conditions.

The histogram model-based experiment were divided in two parts. First, we were experimenting with various training models in order to build a model with the best performance from the find the smallest amount of training data. In the second part of the experiment we were seeking the VAD combination with the smallest average total error rate result. We discovered that the model is intolerable to a training with initially false ground truth. In the experiment all analyzed VAD combinations outperformed standalone AMR1 results and Enegy, G.729 and AMR1 combination showed the smallest total error rate. In the further research of the histogram model-based approach, the temporal context and larger amount of VAD could be used.

Bibliography

[1] R. G. Bachu, S. Kopparthi, B. Adapa, and B.D. Barkana. Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal. In *American Society for Engineering Education (ASEE) Conferense Proceedings*, pages 1–7, 2008.

[2] P.A. Barrett. Information tone handling in the half-rate gsm voice activity detector. In *Communications, 1995. ICC '95 Seattle, 1995 IEEE International Conference on 'Gateway to Globalization'*, volume 1, pages 72 –76 vol.1, June 1995.

[3] A. Benyassine, E. Shlomot, H.-Y. Su, D. Massaloux, C. Lamblin, and J.-P. Petit. Itu-t recommendation g.729 annex b: A silence compression scheme for use with g.729 optimized for v.70 digital simultaneous voice and data applications. *IEEE Communications Magazine*, 35(9):64 –73, September 1997.

[4] S. Boslaugh and P. A. Watters. *Statistics in a Nutshell.* O'Reilly, 2008.

[5] Joon-Hyuk Chang, Nam Soo Kim, and S.K. Mitra. Voice activity detection based on multiple statistical models. *IEEE, Transactions on Signal Processing*, 54(6):1965 – 1976, June 2006.

[6] Shi-Huang Chen and Jhing-Fa Wang. A wavelet-based voice activity detection algorithm in noisy environments. *Electronics, Circuits and Systems*, 3:995–998, 2002.

[7] Texas Instruments Linguistic Data Consortium. Tidigits, 1982. `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S10`.

[8] A. Davis, S. Nordholm, and R. Togneri. Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold. *IEEE Transactions on Audio, Speech and Language Processing*, 14:412–424, March 2006.

[9] P.A. Estevez, N. Becerra-Yoma, N. Boric, and J.A. Ramirez. Genetic programming-based voice activity detection. *Electronics Letters*, 41(20):1141 – 1143, sept. 2005.

[10] G. Evangelopoulos and P. Maragos. Multiband modulation energy tracking for noisy speech detection. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(6):2024 –2038, November 2006.

[11] M. Farsinejad and M. Analoui. A new robust voice activity detection method based on genetic algorithm. In *Australasian Telecommunication Networks and Applications Conference, 2008*, pages 80 –84, December 2008.

[12] D.K. Freeman, G. Cosier, C.B. Southcott, and I. Boyd. The voice activity detector for the pan-european digital cellular mobile telephone service. In *International Conference on Acoustics, Speech, and Signal Processing, 1989. ICASSP-89.*, pages 369 –372 vol.1, may 1989.

[13] Roger L. Freeman. *Fundamentals of Telecommunications.* Wiley-IEEE Press, 2nd edition, 2005.

[14] M. Fujimoto, K. Ishizuka, and T. Nakatani. A voice activity detection based on the adaptive integration of multiple speech features and a signal decision scheme. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.*, pages 4441 –4444, April 2008.

[15] S. Ikram and H. Malik. Digital audio forensics using background noise. In *IEEE International Conference on Multimedia and Expo (ICME), 2010*, pages 106 –110, July 2010.

[16] Tomi Kinnunen. *Spectral Features for Automatic Text-Independent Speaker Recognition.* Licentiate's thesis, University of Joensuu, December 2003.

[17] Tomi Kinnunen, Evgenia Chernenko, Marko Tuononen, Pasi Fränti, and Haizhou Li. Voice activity detection using mfcc features and support vector machine. In *Speech and Computer (SPECOM)*, volume 2, pages 556–561, Moscow, Russia, October 2007.

[18] Ron Kohavi and Foster Provost. Glossary of terms. *Mach. Learn.*, 30(2-3), 1998.

[19] Ludmila I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3:135–148, 2002.

[20] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004.

[21] ELRA (European language resources association). Aurora project database 2.0 - evaluation package, 1993. `http://catalog.elra.info/product_info.php?products_id=693`.

[22] Yuan Liang, Xianglong Liu, Mi Zhou, Yihua Lou, and Baosong Shan. A robust voice activity detector based on weibull and gaussian mixture distribution. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 2, pages V2–26 –V2–30, July 2010.

[23] Kenneth O. May. A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decision. *Econometrica*, 20(4):680–684, 1952.

[24] R. Merris. *Combinatorics*. Wiley, USA, 2nd edition, 2003.

[25] Richard Meston. Sorting through gsm codecs: A tutorial, 2003.

[26] N. Mokhtar, H. Arof, F. R. Mahamd Adikan, and M. Mubin. Real time noise-speech discrimination in time domain for speech recognition application. *Scientific Research and Essays*, 6(1):18–22, 2011.

[27] NIST (National Institute of Standards and Technology). Speaker recognition evaluation, 2005. http://www.itl.nist.gov/iad/mig//tests/spk/2005.

[28] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[29] L. Rabiner, R. Schafer, and W. Ronald. *Digital processing of speech signals*. Prentice-Hall, Englewood Cliffs, N.J. :, 1978.

[30] J. Ramirez, J. M. Gorriz, and J. C. Segura. Voice Activity Detection. Fundamentals and Speech Recognition System Robustness. In *Robust Speech Recognition and Understanding*. InTech, June 2007.

[31] J. Ramirez, J.C. Segura, C. Benitez, A. de la Torre, and A. Rubio. Voice activity detection with noise reduction and long-term spectral divergence estimation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, volume 2, pages ii – 1093–6 vol.2, May 2004.

[32] J. Ramirez, J.C. Segura, C. Benitez, A. de la Torre, and A. Rubio. An effective subband osf-based vad with noise reduction for robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(6):1119 – 1129, nov. 2005.

[33] Javier Ramírez, José C. Segura, Carmen Benítez, Ángel de la Torre, and Antonio Rubio. Efficient voice activity detection algorithms using long-term speech information. *Speech Communication*, 42(3-4):271–287, April 2004.

[34] P. Renevey and A. Drygajlo. Entropy Based Voice Activity Detection in Very Noisy Conditions. In *Proceedings of 7th European Conference on Speech Communication and Technology, (EUROSPEECH'2001)*, pages pp. 1887–1890, 2001.

[35] Jongseo S., Nam S. K., and Wonyong S. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1 –3, January 1999.

[36] European Standard (Telecommunications series). Voice activity detector (vad) for adaptive multi-rate (amr) speech traffic channels, 1999. ETSI EN 301 708 v7.11 standard description.

[37] European Standard (Telecommunications series). Speech processing, transmission and quality aspects. advanced front-end feature extraction algorithm, 2007. ETSI ES 202 050 standard description.

[38] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

[39] Skype. Silk speech codec, 2009. Accessed 14 April 2011; http://developer.skype.com/silk.

[40] J. Soumagne, J.-P. Adoul, and S. Morissette. A new concept for encoding speech amplitude time quantization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '84).*, volume 9, pages 428 – 431, March 1984.

[41] Ivan J. Tashev. *Sound Capture and Processing: Practical Approaches.* Wiley, 2009.

[42] Marko Tuononen, Rosa Gonzalez Hautamäki, and Pasi Fränti. Automatic voice activity detection in different speech applications. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*, (e-Forensics '08), pages 12:1–12:6, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[43] M. Turunen, J. Hakulinen, K.-J. Räihä, E.-P. Salonen, A. Kainulainen, and P. Prusi. An architecture and applications for speech-based accessibility systems. *IBM Systems Journal*, 44:485–504, August 2005.

[44] Saeed V. Vaseghi. *Advanced digital signal processing and noise reduction.* Wiley, 2008.

[45] Andrey R. Webb. *Statistical pattern recognition.* Wiley, USA, 2nd edition, 2002.

Appendix A

Energy and Entropy VADs source code

```
% Initialization
FrameLen = 240;
FrameShift = FrameLen / 3;
W = hamming(FrameLen);

% normalize the signal
s = s / max(abs(s));
% divide the signal into overlapping frames
Frames = enframe(s, W, FrameShift);

% call energy or entropy VAD
vad = VoiceActivityDetector(Frames);


% Energy VAD
function indic = VoiceActivityDetector(Frames)
```

```
S = 20*log10(std(Frames') + eps);
max1 = max(S);
indic = (S>max1-30) & (S>-55);
%%


% Entropy VAD
function indic = VoiceActivityDetector(frames)
NFFT = 512;

nframes = size(frames, 1);
spec = fft(frames, NFFT, 2);
H = zeros(nframes, 1);

for i = 1:nframes
    spec_frame = spec(i,:);
    p_sum = sum(abs(spec_frame));
    p = abs(spec_frame).^2 / p_sum;
    h = -sum(p.*log(p));
    H(i) = h;
end

min1 = min(H(H > 0));
H(H ≤ 0) = min1;

std1 = std(H);
mean1 = mean(H);
indic = (H > 0.4) & (H > min1 + abs(mean1 - std1));
```