INTERSPEECH 2009 BRIGHTON

Fusing Fast Algorithms to Achieve Efficient Speech Detection in FM Broadcasts

Stéphane Pigeon, Patrick Verlinde

Royal Military Academy, Brussels, Belgium

Stephane.Pigeon@rma.ac.be, Patrick.Verlinde@rma.ac.be

Abstract

This paper describes a system aimed at detecting speech segments in FM broadcasts. To achieve high processing speeds, simple but fast algorithms are used. To output robust decisions, a combination of many different algorithms has been considered. The system is fully operational in the context of Open Source Intelligence, since 2007.

Index Terms: speech detection, speech segmentation, fusion, intelligence.

1. Introduction

Open Source Intelligence (OSINT) consists of acquiring information from publicly available sources. Among those sources, FM broadcasts are monitored and relevant speech is analyzed. In such a context, the Royal Military Academy developed a convenient interface to help operators in localizing and auditioning spoken segments in FM recordings. Relying on the same algorithms, we designed an audio plug-in which automatically removes non-spoken segments from radio recordings without any human intervention. This plug-in runs fast - about 100 times faster than real-time - and has been proven robust against various noise conditions and languages.

This paper is organized as follows. Section 2 provides the reader with an overview of our system. Section 3 details the individual algorithms working with audio as an input, the so-called *Audio Experts* (AEs). Section 4 describes how these AEs are optimally combined during a process referred as *Fusion*. Section 5 presents the application's user interface and the performance achieved by the automatic plug-in.

2. Overview and Terminology

The goal of our system is to segment relevant speech from all other audio signals. What makes speech relevant depends on a particular OSINT context. Most of the time, relevance is associated to monologs, dialogs and interviews but excludes speech as found in advertisements. Relevant speech can sometimes be sung (preaches) or may be layered on a musical background. From now on, we will use the word *Speech* to designate *relevant* speech only, while *Garbage* will refer to anything else, mainly music in our case. What makes the classification tricky is the small difference which sometimes exists between those two classes. For example, spoken words uttered on the top of a musical background (Speech) and a singer's a capella passage in a song (Garbage) do not belong to the same class. As long as an audible difference exists for a human listener, our system should be capable of discriminating the speech and garbage classes.

FM recordings have been sampled at 22.05 kHz (half the audio CD standard) in 16-bit, uncompressed format. Often, lower sampling frequencies are considered, 8 kHz being a reference in the speech community. Working with a higher sampling frequency is important as significant differences between the Speech and Garbage classes can be found above 4kHz.

Decisions whether an audio signal belongs to Speech or Garbage are taken over regularly spaced, non-overlapping, time intervals referred to as *Chunks*. Chunk length has been fixed to 5 seconds. We found this setting a minimum to reliably detect rhythm in a song (see our AEs in next section). We did not consider longer intervals as not to misclassify short passages of relevant speech confined between garbage chunks. Since classification is performed at chunk level, the exact position where speech occurs is not detected. A *Segment* designates a number of contiguous chunks belonging to a same class. In order not to miss any word, speech segments delivered to the operator are extended by one garbage chunk in both directions. Our system thus detects speech with a precision of 5 seconds, a constraint which was not felt being a hassle for human operators.

3. Audio Experts

Conceptually, AEs represent the core of our system. With an audio chunk as an input, each AE outputs a scalar metric in relationship with the amount of speech present in the chunk. In practice though, a supplemental layer is inserted between the chunk and the AEs. This layer consists of chunk parameters shared among AEs. In such a perspective, AEs represent different derivations of the common set of chunk parameters. This section first describes the common chunk parameters, then how AEs are derived from them.

3.1. Chunk Parameter Layer

Let c_l denote the chunk under analysis with l its length (in number of samples). As most of the analysis will be performed on a frame basis, let m denote the number of samples in a frame and n the number of frames in a chunk. We used the standard duration of 30 ms for a frame length. By rounding m, n and l to convenient values, we ended up using $(m;n;l)=(660;167;110220)$. The chunk parameter layer is build by computing various quantities inspired from [1][2][3] in the order described below:

- Squared chunk $s_l=c_l^2$
- Average energy $E^{avg}=mean(s_l)$
- Local energy e_l as a moving average of m consecutive s_l samples
- The variance of the local energy $V=var(e_l)$
- The average lowest local energy E^{avg_min} obtained by computing the lowest energy every 0.5s (i.e. every tenth of a chunk) then averaging the results across the chunk
- Percentage of local energy samples below half of the average energy $P=\%[e_l<(E^{avg}/2)]$

- The 5 most energetic frames are isolated from c_l by looking for maxima in e_l. These frames will be referred to as *Top5* frames. By restricting subsequent processing to those frames, a factor 30 in speed increase is achieved.
- Strength of the fundamental frequency f_0 in the human range, denoted as S_{f0}. For every Top5 frame, we isolate the maximum value taken by the frame's autocorrelation between 75 Hz and 350 Hz. S_{f0} is obtained by averaging those 5 values.
- Bass content below human range, *B*. The ratio between the energy below human voice frequency range (below 100 Hz) and the energy found in the voice band (between 200Hz and 4000 Hz) is computed for each Top5 frame. B is obtained as an average over those 5 frames.
- Attack density, δ. By subtracting each chunk sample with its neighbor then taking the absolute value of those differences, we construct $\Delta_{(l-1)}$ the delta chunk vector. The attack density is computed as the percentage of delta chunk samples greater than twice their average.
- Rhythm strength, *R*. The computation of the rhythmic strength is in a way similar to the computation of S_{f0} but performed in the much lower frequency domain and over a longer period of time. By introducing a delay of a chunk in the processing chain, the previous, the current and the next local energy vectors e_l are concatenated then down-sampled by a factor *m*. The resulting samples are differentiated to put emphasis on sound attacks where rhythm occurs. Given the median filter used to derive local energies, the best transient detection is achieved with a delta of 2 (each sample is subtracted from the sample next to its neighbor). Finally, the autocorrelation of the resulting signal is computed and *R* is given as its maximum found between 5 and 120 bpm (beats-per-minute).

3.2. Audio Experts

From the previous measurements, eight AEs are derived. These have been categorized in three families: energy-based, frequency-based and time-based experts.

3.2.1. Energy-based AEs

- $AE_1 = E^{avg}$
- $AE_2 = V$
- $AE_3 = P$
- $AE_4 = E^{avg\_min}/E^{avg}$

Our first expert AE_1 simply outputs the average energy over a chunk. The average energy is however hardly found as a discriminative criterion in the literature. The reason why is obvious. Speech can be loud or quiet, and so does garbage too. In such a context, loudness on its own - or the average chunk energy here - cannot offer any discriminative power. In the scope of our particular application however, loudness turned to be much more effective than expected, as will be explained now. Offering a "signature" sound is of importance for a radio station: not only to be identified from other broadcasters, but to tune in with its targeted audience. This distinctive sound is achieved by using various audio processors at the end of the audio path and compressors in particular. By reducing the dynamic range, compressors help making loudness uniform across different programs, different songs or between a speaker's voice and music. Initially compressors were mostly used as specialized automatic gain controllers (AGC). With the advent of multi-band compressors and the inclusion of sound parameters that go beyond simple AGC, present-day compressors literally "sculpt" the sound of a radio channel : making it perceptually louder than the next station and more aggressive, or - at the opposite - relaxing and airy. The leveling carried out automatically and consistently by compressors, helps a lot to turn speech segments into a perceptually even ensemble, and music into another. This is the reason why an average energy can become a discriminative parameter. In general, garbage chunks exhibit a greater energy than speech. But sometimes, the opposite occurs as in the case of news-oriented stations, where one wants the voice of the speaker to call for attention and the musical interludes to sit in the background. Three important constraints have to be fulfilled when using AE_1 as an expert. First, as average speech and garbage levels are radio-dependent, radio stations have to be individually trained. Second, training should be performed over a long period of time if one wants to achieve optimal results (a day minimum, one week best) : the difference induced by a given compressor on the speech and garbage classes can only be extracted from a large amount of varied voiced and music material. Last but not least, recording levels should be fixed once for all. Any small discrepancy in the recording levels between training and operation phases may severely bias the performance of AE_1.

Our second expert AE_2 refers to the variability of the local energy in a chunk. This variability differs from speech and garbage chunks and the amount of difference depends again on a station's audio path settings. Therefore - for this criterion to offer a discriminative power - the same considerations as for AE_1 apply (the individual trainings and the fixed recording levels).

Our third expert AE_3 is related to the amount of micro-gaps in the signal. Unlike our first two experts, this criterion is commonly found in speech segmentation literature and indeed works extremely well with clean recordings. Due to its intrinsic nature, speech contains a lot of breaks in the sound : one may intuitively think of the speaker breathing, but it actually relates to those imperceptible but measurable micro-gaps intrinsically derived from speech articulation (think about producing a plosive without any break in sound right before). These gaps are typically 30ms long and do not exist in music material since as with the number of instruments playing, a musical signal tends to be represented by an uninterrupted flow. AE_3 thus exhibits much higher values for speech than for garbage chunks.

AE_4 compares the average energy of the lowest energetic frames, with the average energy over a chunk. This information complements AE_3 which only provides information about the number of low energetic frames in a chunk, not their actual level.

3.2.2. Frequency-based AEs

- $AE_5 = S_{f0}$
- $AE_6 = B$

AE_5 relates to the fundamental frequency strength in the human voice frequency range. This expert thus takes higher values for speech, but for music too when the bass plays in

the lower vocal frequency range. Still, AE_5 is discriminative as when it gets a low value, speech can be excluded.
AE_6 compares the frequency content below human range with the energy found in the human voice band. Unlike phone-quality material, music as broadcasted through FM exhibits a stronger low end as compared to voice.

3.2.3. Time-based AEs

- $AE_7 = \delta$
- $AE_8 = R$

Music is characterized by the presence of a rhythm, often accentuated by percussive instruments. AE_7 relates to this percussiveness while AE_8 measures the strength of the rhythm when present. AE_8 is possibly the most interesting expert in our application, as it is the only one to catch the difference between a speaking and a singing voice : the singer sticks to a given musical tempo and note quantization, not a speaker.

4. Fusion

Each expert thus assigns a single metric to each chunk. Some experts will associate higher metrics to speech chunks, others will do the opposite. This discrepancy is not important, only consistency matters. Fusion refers to the process of aggregating these individual metrics into a robust score. To learn each expert's behavior, performance and reliability, this fusion process has to undergo a training phase first. Only then, the algorithm will be able to optimally combine the different metrics in operational conditions. Linear Discriminant Analysis (LDA) [4] has been selected as a fusion algorithm.

4.1. Linear Discriminant Analysis

Let $z = [z_1\ z_2\ ...\ z_N]$ denote the metric vector obtained by concatenating all individual expert metrics z_i given on a particular chunk, $i=1..N$ with N being the total number of experts available. Let $T(z|c)$ denote the metric distribution conditionally to the chunk class $c=\{s(peech),g(arbage)\}$. To optimally determine whether a chunk is speech or garbage, we use the likelihood ratio:

$$\frac{T(z|s)}{T(z|g)} \tag{1}$$

The chunk will be classified as speech when this ratio gets higher than a given threshold. $T(z|s)$ and $T(z|g)$ are unknowns. A common hypothesis consists in approximating these distributions by Gaussians

$$f_C(z) = (2\pi)^{-N/2} \exp\left\{-\frac{1}{2}(z-\mu_C)'\Sigma^{-1}(z-\mu_C)\right\} \tag{2}$$

where μ_c represents the average expert metric vector within a given class c and Σ the covariance matrix between experts. μ_s, μ_g and Σ can be estimated from the metrics obtained during training time. Let x denote the n_s metric vectors related to the speech training chunks and y the n_g metric vectors related to the garbage chunks.

$$\hat{\mu}_s = \sum_{q=1}^{n_s} x_q / n_s \tag{3}$$

$$\hat{\mu}_g = \sum_{q=1}^{n_g} y_q / n_g \tag{4}$$

$$\hat{\Sigma}_s = \sum_{q=1}^{n_s} (y_q - \hat{\mu}_s)(y_q - \hat{\mu}_s)'/(n_s - 1) \tag{5}$$

$$\hat{\Sigma}_g = \sum_{q=1}^{n_g} (y_q - \hat{\mu}_g)(y_q - \hat{\mu}_g)'/(n_g - 1) \tag{6}$$

$$\hat{\Sigma} = \left[(n_s - 1)\hat{\Sigma}_s + (n_g - 1)\hat{\Sigma}_g\right]/(n_s + n_g - 2) \tag{7}$$

By substituting equations (2) to (7) into equation (1), one obtains the linear discriminant function:

$$D_L(z) = (z - \frac{1}{2}(\hat{\mu}_s + \hat{\mu}_g))'\hat{\Sigma}^{-1}(\hat{\mu}_s - \hat{\mu}_g) \tag{8}$$

The value of this discriminant function is computed for every chunk belonging to the training set. Then, one sets a threshold which best separates speech and garbage classes. The classification of a test segment is obtained by comparing $D_L(z)$ with the threshold found earlier and it will be associated with the speech class, when it falls above.

4.2. Performance

In such a binary classification problem, two kinds of errors may occur: rejecting speech chunks (i.e. classifying speech as garbage) and accepting garbage chunks inside speech segments. Those errors are commonly referred as *False Rejection* (FR) and *False Acceptance* (FA) respectively. The tradeoff between those two types of errors directly relates to the value of the acceptance threshold applied to equation (8). By continuously varying this threshold, one generates a curve in FA/FR space which represents the overall performance of the system in terms of possible FA/FR tradeoffs. The *Equal Error Rate* (EER) refers to the operating point where FA=FR (=EER).

To rate our algorithm performance, we used a major commercial radio station, recorded from six o' clock in the morning till midnight, for two consecutive weekdays (2x18h, 22.05kHz, 16-bit, monaural). Day one was used as a training set, day two as a test set. These two sets have been manually cleaned from all audio segments which couldn't be clearly labeled either as speech or garbage: think of spoken advertisements which may be considered as speech sound-wise, but garbage if one considers their contents and our application in mind. After the cleaning up, we came up with the speech and garbage durations listed in Table 1. As training and test files have been acquired across the same period of time, on the same channel, they roughly contain the same proportion of speech and garbage (60% of speech for 40% of music approximately). The training file thus preserves the station's average a priori speech probability. Feeding representative *a priori* probabilities to the fusion algorithm is important and ensures that the operating point selected along the performance curve at training time translates properly during test time or when the system is operational.

Table 1 : *Training and Test contents*

	Speech	Garbage	Total
Train	7h44m03s	5h42m34s	13h26m37s
Test	8h10m56s	5h29m21s	13h40m17s

Speech segments contain some challenging segments as music often appears in the background, sometimes imprinted with a strong rhythmic pattern. There is also a long coverage of a soccer game - live from the stadium, with a lot of background noise (crowd) - found in the test set but absent from the training set.

Feeding the training files to our fusion algorithm achieved an outstanding equal error rate of 0.62%. Under test, this operating point shifted to $\{FA;FR\}_{test}$={0.91%;0.96%} which is still remarkable. When the acceptance threshold is set such as to reject no speech at training time ($FR_{train}=0$), 95.53% of the garbage chunks are properly filtered out (FA_{train}=4.47%). During the test, this rate decreased to 92.73% (FA_{test}=7.27% with FR_{test} still null). Figure 1 provides a continuous performance curve and a few possible {*FA;FR*} tradeoffs one can achieve around the EER.

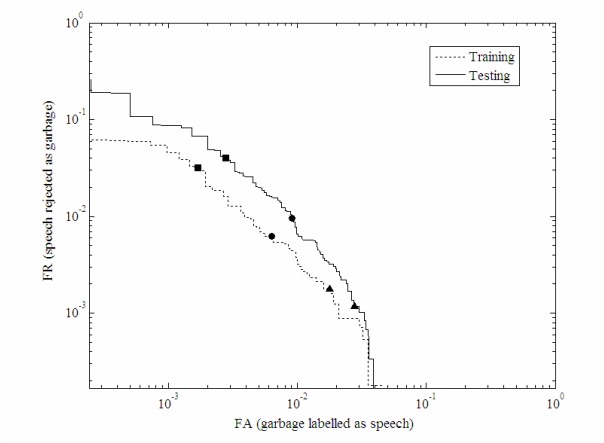

Figure 1 : Performance achieved at Training (dotted) and Test (solid) times and 3 possible operating points (the equal error rate as a circle). A logarithmic scale has been used as to best separate the performance curves from the axes' origin.

It is of importance to remember that this level of performance is achieved when classifying each 5s chunk independently from each other. In practice, garbage and speech segments extend much over the duration of a chunk. Isolated chunks are not likely to happen unless misclassified. Therefore, by filtering out isolated chunks, one further improves the overall performance of the system beyond the *FA* and *FR* rates mentioned earlier.

Let us stress again that this outstanding level of performance can be only achieved when radio stations are trained individually, as some experts depend heavily on the nature of the audio post-procsessing a radio station applies to shape its sound.

Automatic classification is only achieved when our segmentation system runs as an audio plug-in without any human supervision. Relying on the same algorithms, we also developped a grapical tool which represents the audio as a coloured bar. Colours are selected to intuitively represent the values of $D_L(z)$, with lower values in red (garbage), higher in green (speech) and all intermediate colors in between. The operator easily locates speech segments by picking up the green colored areas and previews or saves them by simply clicking those areas. Such a bar representation is illustrated in Figure 2.

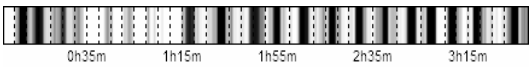

Figure 2 : A color-coded audio bargraph, as a compact representation of broadcasted contents. Speech areas are represented in green (or white in this B&W print), garbage areas in red (black). Blue areas (grey) are equally associated with advertisements or vocal solos in music segments.

5. Conclusions

This paper described an operational system to segment speech from FM broadcasts, with an intelligence application in mind. A high performance has been achieved when training radio stations individually. The system is fully operational since 2007. Future work focuses on detecting particular sounds to trigger subsequent recordings (a news jingle to force the recording of the upcoming news) or - on the opposite - remove part of the recording (a spoken advertisement, which too often gets segmented as useful speech while not being pertinent).

6. References

[1] Eric Scheirer and Malcolm Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator", in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 1997.

[2] Enric Guaus and Eloi Batlle, "A non-linear rhythm-based style classification for Broadcast Speech-Music Discrimination", in Proceedings of 116th AES Audio Engineering Society Convention, 2004.

[3] George Tzanetakis, Georg Essl and Perry Cook,, "Automatic Musical Genre Classification of Audio Signals", in Proceedings of the Int. Symposium on Music Information Retrieval ISMIR 2001.

[4] G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition. Wiley, 1992.