

A multi-class method for detecting audio events in news broadcasts

Sergios Petridis, Theodoros Giannakopoulos, and Stavros Perantonis

Computational Intelligence Laboratory,
Institute of Informatics and Telecommunications,
National Center of Scientific Research Demokritos
petridis@iit.demokritos.gr, tyiannak@gmail.com, sper@iit.demokritos.gr

Abstract. In this paper we propose a method for audio event detection in video streams from news. Apart from detecting speech, which is obviously the major class in such content, the proposed method detects five non-speech audio classes. The major difficulty of the particular task lies in the fact that most of the audio events (apart from speech) are actually background sounds, with speech as the primary sound. We have adopted a set of 21 statistics computed on a mid-term basis over 7 audio features. A variation of the One Vs All classification architecture has been adopted and each binary classification problem is modeled using a separate probabilistic Support Vector Machine. For evaluating the overall method, we have defined the precision and recall rates for the event detection problem. Experiments have shown that the proposed method can achieve high precision rates for most of the audio events of interest.

Key words: Audio event detection, Support Vector Machines, Semi-automatic multimedia annotation

1 Introduction

With the huge increase of multimedia content that is made available over Internet during the last years, a number of methods have been proposed for automatic characterization of this content. Especially for the case of multimedia files from **news** broadcasts, the usefulness of an automatic content recognition method is obvious. Several methods have been proposed for automatic annotation of news videos, though, only a few of those make extensive use of the audio domain ([1], [2], [3]). In this work, we propose an algorithm for event detection in real broadcaster videos, that is based only on the audio information. This work is part of the CASAM European project (www.casam-project.eu), which aims at computer-aided semantic annotation (i.e., semi-automatic annotation) of multimedia data. Our main goal is to detect (apart from speech) five non-speech sounds that were met in our datasets from real broadcasts. Most of these audio events were secondary (background) sounds to the main event which is obviously speech. This task of recognizing background audio events in news can help in extracting richer semantic information from such content.

2 Audio class description

Since the purpose of this work is to analyze audio streams from news, it is expected that the vast majority of the audio data is speech. Therefore, the first of the audio class we have selected to detect is **speech**. Speech tracking may be useful if its results were used by another audio analysis module, e.g. by a speech recognition task. Though, the detection of speech as an event is not of major importance in a news audio stream. Therefore, the following more semantically rich audio classes have also been selected: **music**, **sound of water**, **sound of air**, **engine sounds** and **applause**. In a news audio stream the above events most of the times exist as background events, with speech being the major sound. Hence, the detection of such events is obviously a hard task. It has to be noted that an audio segment can, at the same time, be labeled as speech and as some other type of event, e.g. music. This means that a segment contains speech and background music.

3 Audio Feature Extraction

3.1 Short-term processing for audio feature extraction

Let $x(n), n = 1, \dots, L$, be the audio signal samples and L the signal length. In order to calculate any audio feature of x , it is needed to adopt a *short-term processing technique*. Therefore, the audio signal is divided in (overlapping or non-overlapping) short-term windows (frames) and the feature calculation is executed for each frame. The selection of the window size (and corresponding step) is sometimes crucial for the audio analysis task. The window size should be large enough for the feature calculation stage to have enough data. On the other hand, it should be short enough for the (approximate) stationarity to be valid. Common window sizes vary from 10 to 50 msecs, while the window step is associated to the level of overlap. If, for example, 75% of overlap is needed, and the window size is 40 msecs, then the window step is 10 msecs.

As long as the window size and step is selected the feature value f is calculated *for each frame*. Therefore, an M -element array of feature values $\mathbf{F} = f_j, j = 1, \dots, M$, for the whole audio signal is calculated. Obviously, the length of that array is equal to the number of frames: $M = \lfloor \frac{L-S}{N} \rfloor + 1$, where: N the window length (number of samples), S the window step and L the total number of audio samples of the signal.

3.2 Mid-term processing for audio feature extraction

The process of short-term windowing, described in Section 3.1, leads, for each audio signal, to a sequence \mathbf{F} of feature values. This sequence can be used for processing / analysis of the audio data. Though, a common technique is the processing of the feature *in a mid-term basis*. According to this technique, the audio signal is first divided into mid-term windows (segments) and then for each

segment the short-term process is executed. In the sequel, the sequence \mathbf{F} , which has been extracted for each segment, is used for calculating a statistic, e.g., the average value. So finally, each segment is represented by a single value which is the statistic of the respective feature sequence. Common durations of the mid-term windows are 1 – 10 secs. We have chosen to use a 2 second mid-term window, with a 1 second step (50% step). This particular window length was chosen in order to use a window that contains a statistically sufficient number of short-term windows. On the other hand, the adopted mid-term step provides a satisfactory time resolution of the returned results.

3.3 Adopted Audio Features and Respective Statistics

We have implemented 7 audio features, while, **for each feature three statistics have been used in a mid-term basis: mean value, standard deviation and std by mean ratio.** Therefore, in total, **each mid-term window is represented by 21 feature values.** In the following, the 7 features are presented, along with some examples of their statistics for different audio classes.

Energy Let $x_i(n), n = 1, \dots, N$ the audio samples of the i -th frame, of length N . Then, for each frame i the energy is calculated according to the equation: $E(i) = \frac{1}{N} \sum_{n=1}^N |x_i(n)|^2$. This simple feature can be used for detecting silent periods in audio signals, but also for discriminating between audio classes. The variations in the speech segments are usually higher than in music. This is a general observation and it has a physical meaning, since speech signals have many silence intervals between high energy values, i.e., the energy envelope alternates rapidly between high and low energy states. Therefore, a statistic that can be used for the case of discriminating signals with large energy variations (like speech, gunshots etc.) is the standard deviation σ^2 of the energy sequence. In order to achieve energy-independence, the standard deviation by mean ratio ($\frac{\sigma^2}{\mu}$) has also been used ([4]).

Zero Crossing Rate Zero Crossing Rate (ZCR) is the rate of sign-changes of a signal, i.e., the number of times the signal changes from positive to negative or back, per time unit. It is defined according to the equation: $Z(i) = \frac{1}{2N} \sum_{n=1}^N |sgn[x_i(n)] - sgn[x_i(n-1)]|$, where $sgn(\cdot)$ is the sign function. This feature is actually a measure of noisiness of the signal. Therefore, it can be used for discriminating noisy environmental sounds, e.g., rain. Furthermore, in speech signals, the $\frac{\sigma^2}{\mu}$ ratio of the ZCR sequence is high, since speech contains unvoiced (noisy) and voiced parts and therefore the ZCR values have abrupt changes. On the other hand, music, being largely tonal in nature, does not show abrupt changes of the ZCR. ZCR has been used for speech-music discrimination ([5], [4]) and for musical genre classification ([6]).

Energy Entropy This feature is a measure of abrupt changes in the energy level of an audio signal. It is computed by further dividing each frame into K sub-frames of fixed duration. For each sub-frame j , the normalized energy e_j^2 is calculated, i.e., the sub-frame’s energy, divided by the whole frame’s energy: $e_j^2 = \frac{E_{subFrame_j}}{E_{shortFrame_i}}$. Therefore e_j is a sequence of normalized sub-frame energy values, and it is computed for each frame. Afterwards, the entropy of this sequence is computed using the equation: $H(i) = -\sum_{j=1}^K e_j^2 \cdot \log_2(e_j^2)$.

The entropy of energy of an audio frame is lower if there are abrupt changes present in that audio frame. Therefore, it can be used for discrimination of abrupt energy changes, e.g. gunshots, abrupt environmental sounds, etc.. In Figure 1 an example of an Energy Entropy sequence is presented for an audio stream that contains: classical music, gunshots, speech and punk-rock music. Also, the selected statistics for this example are the maximum value and the $\frac{\sigma^2}{\mu}$ ratio. It can be seen that the minimum value of the energy entropy sequence is lower for gunshots and speech.

Fig. 1. Example of Energy Entropy sequence for an audio signal that contains four successive homogeneous segments: classical music, gunshots, speech and punk-rock music

Spectral Centroid Frequency domain (spectral) features use as basis the Short-time Fourier Transform (STFT) of the audio signal. Let $X_i(k)$, $k = 1 \dots, N$, be the Discrete Fourier Transform (DFT) coefficients of the i -th short-term frame, where N is the frame length. The spectral centroid, is the first of the spectral domain features adopted in the CASAM audio module. The spectral centroid C_i , of the i -th frame is defined as the center of “gravity” of its spectrum, i.e., $C_i = \frac{\sum_{k=1}^N (k+1)X_i(k)}{\sum_{k=1}^N X_i(k)}$. This feature is a measure of the spectral position, with high values corresponding to “brighter” sounds.

Position of the Maximum FFT Coefficient This feature directly uses the FFT coefficients of the audio segment: the position of the maximum FFT coefficient is computed and then normalized by the sampling frequency. This feature is another measure of the spectral position.

Spectral Rolloff Spectral Rolloff is the frequency below which certain percentage (usually around 90%) of the magnitude distribution of the spectrum is concentrated. This feature is defined as follow: if the m -th DFT coefficient corresponds to the spectral rolloff of the i -th frame, then the following equation holds: $\sum_{k=1}^m X_i(k) = C \sum_{k=1}^N X_i(k)$, where C is the adopted percentage. It has to be noted that the spectral rolloff frequency is normalized by N , in order to achieve values between 0 and 1. Spectral rolloff is a measure of the spectral shape

of an audio signal and it can be used for discriminating between voiced and unvoiced speech ([7], [8]). In Figure 2, an example of a spectral rolloff sequence is presented, for an audio stream that contains three parts: music, speech and environmental noise. The mean and the median values of the spectral sequence for each part of the audio streams are also presented. It can be seen that both statistics are lower for the music part, while for the case of the environmental noise they are significantly higher.

Fig. 2. Example of a spectral rolloff sequence for an audio signal that contains music and speech and environmental noise.

Spectral Entropy Spectral entropy ([9]) is computed by dividing the spectrum of the short-term frame into L sub-bands (bins). The energy E_f of the f -th sub-band, $f = 0, \dots, L - 1$, is then normalized by the total spectral energy, yielding $n_f = \frac{E_f}{\sum_{f=0}^{L-1} E_f}$, $f = 0, \dots, L - 1$. The entropy of the normalized spectral energy n is then computed by the equation: $H = -\sum_{f=0}^{L-1} n_f \cdot \log_2(n_f)$.

In Figure 3 an example of the spectral entropy sequence is presented, for an audio stream that contains a speech and a music part. It is obvious that the variations in the music part are significantly lower. A variant of the spectral entropy called “chromatic entropy” has been used in [10] and [11] in order to discriminate in an efficient way speech from music.

Fig. 3. Example of Spectral Entropy sequence for an audio stream that contains a speech and a music segment

4 Event Detection

As described in Section 3, the mid-term analysis procedure leads to a vector of 21 elements for each mid-term window. Furthermore, since the selected mid-term window step was selected to be equal to 1 sec, the 21-element feature vector finally represents a 1-sec audio segment from the audio stream. In order to classify each audio segment, we have adopted Support Vector Machines (SVMs) and a variation of the One Vs All classification architecture. In particular, each binary classification task ,e.g., ‘Speech Vs Non-Speech’, ‘Music Vs Non-Music’, etc, is modeled using a separate SVM. The SVM has a soft output which is an estimation of the probability that the input sample (i.e. audio segment) belongs to the

respective class. Therefore, for each audio segment the following soft classification outputs are extracted: $P_{speech}, P_{music}, P_{air}, P_{water}, P_{engine}, P_{applause}$. Furthermore, six corresponding thresholds are defined ($T_{speech}, T_{music}, T_{air}, T_{water}, T_{engine}, T_{applause}$) for each binary classification task. In the training stage, apart from the training of the SVMs, a cross-validation procedure is executed for each of the binary classification sub-problems, in order to estimate the thresholds which maximize the respective **precision** rates. This cross-validation procedure is carried out on each of the binary sub-problems and not in the multi-class problem and the classification decision is based on the respective thresholding criterion.

Before proceeding, it has to be emphasized, that for each audio segment the following three possible classification decisions can exist:

- The label Speech can be given to the segment.
- Any of the non-speech labels can be given to the segment.
- The labels Speech **and** any of the other labels can be given to the segment.
- The segment can be left unlabeled.

Therefore, each audio segment can have from 0 up to 2 content labels, while if the number of labels is 2, then speech has to be one of them. The above assumption stems from the content under study, as explained in Section 2.

In the event detection testing stage, given the 6 soft decisions from the respective binary classification tasks, for each 1-sec audio segment the following process is executed:

- If $P_{speech} \geq T_{speech}$, then the label 'Speech' is given to the segment.
- For each of the other labels $i, i \in \{music, air, water, engine, applause\}$: if $P_i < T_i$ then $P_i = 0$.
- Find the maximum of the non-speech soft outputs and its label $imax$.
- If $P_{imax} > T_{imax}$ then label the segment as $imax$.

The above process is repeated for all mid-term segments of the audio stream. As a final step, successive audio segments that share the same label are merged. This leads to a sequence of audio events, each one of which is characterized by its label and its time limits.

5 Experimental Results

5.1 Datasets and manual annotation

For training - testing purposes, two datasets have been populated in the CASAM project: one from the a German international broadcaster (DW - Deutsche Welle) and the second from the Portugese broadcaster (Lusa - Agncia de Noticias de Portuga). Almost 100 multimedia streams from the above datasets have been manually annotated, using the Transcriber Tool (<http://trans.sourceforge.net/>). The total duration of the multimedia files exceeds 7 hours. The annotation was organized as follow:

- For each speaker in an audio stream, a separate xml object is defined with attributes such as: ID, genre, dialect, etc.
- The annotation on the audio stream is carried out in a segment basis, i.e., audio segments of homogenous content are defined. For each homogenous segment, two labels are also defined: the *primary label* corresponds to the respective speaker ID (e.g., spk1, spk2, etc), while the *secondary label* is related to the type of background sound (e.g., ambient noise, sound of engine, water, wind, etc). It has to be noted that if the segment is a non-speech segment then the primary label is “none”.

In Table 1, a representation for an example of an annotated audio file is shown.

Segment Start	Segment End	Primary Label	Secondary Label
0	1.2	spk1	engine
1.2	3.3	none	engine
3.3	9.8	none	music
9.8	11.5	spk1	water
11.5	16.2	spk2	water
16.2	19.9	spk1	water
...

Table 1. Representation example for an annotated audio file. For each homogenous segment, its limits (start and end) and its primary and secondary labels are defined.

5.2 Method evaluation

Performance measures The audio event detection performance measures (in particular: precision and recall) should differ from the standard definitions used in the classification case. In order to proceed, let us first define an event, as the association of a *segment* s with an element c of a *class set*: $e = \{s \mapsto c\}$. Furthermore, let

- S be the set of all segments of events known to hold as ground truth,
- S' be the set of all segments of events found by the system.

For a particular class label c , let also:

- $S(c) = \{s \in S : s \mapsto c\}$ the set of all ground truth segments associated to class c .
- $\bar{S}(c) = \{s \in S : s \mapsto c' \neq c\}$ the set of all ground truth segments not associated to class c .
- $S'(c) = \{s' \in S' : s' \mapsto c\}$ the set of all system segments associated to class c .

- $\bar{S}'(c) = \{s' \in S' : s' \mapsto c' \neq c\}$ the set of all system segments not associated to class c .

In the sequel let, two segments and a threshold value $t \in (0, 1)$. We define the segment matching function $g : S \times S' \rightarrow \{0, 1\}$ as: $g_t(s, s') = \frac{|s \cap s'|}{|s \cup s'|} > t$. For defining the recall rate, let $A(c)$ be the ground truth segments $s \mapsto c$ for which there exist a matching segment $s' \mapsto c$ $A(c) = \{s \in S(c), \exists s' \in S'(c) : g_t(s, s') = 1\}$. Then, the *recall of class c* is defined as:

$$Recall(c) = \frac{|A(c)|}{|S(c)|} \quad (1)$$

In order to define the event detection precision, let $A'(c)$ be the system segments $s' \mapsto c$ for which there exist a matching segment $s \mapsto c$: $A'(c) = \{s' \in S'(c), \exists s \in S(c) : g_t(s, s') = 1\}$. Then the *precision of class c* is defined as:

$$Precision(c) = \frac{|A'(c)|}{|S'(c)|} \quad (2)$$

Performance results In Table 2, the results of the event detection process is presented. It can be seen that for all audio event types the precision rate is at above 80%. Furthermore, the average performance measures for all non-speech events has been calculated. In particular, the recall rate was found equal to 45%, while precision was 86%. This actually means that almost half of the manually annotated audio events were successfully detected, while 86% of the detected events were correctly classified.

Class names	Recall(%)	Precision(%)
Speech	82	90
SoundofAir	20	82
CarEngine	42	87
Water	52	90
Music	56	85
Applause	59	99
Average (non-speech events)	45	86

Table 2. Detection performance measures

6 Conclusions

We have presented a method for automatic audio event detection in news videos. Apart from detecting speech, which is obviously the most dominant class in the particular content, we have trained classifiers for detecting five other types of

sounds, which can provide important content information. Our major purpose was to achieve high precision rates. The experimental results, carried out over a large dataset from real news streams, indicate that the precision rates are always above 80%. Finally, the proposed method managed to detect almost 50% of all the manually annotated non-speech events, while from all the detected events 86% were correct. This is a rather high performance, if we take into consideration that most of these events exist as background sounds to speech in the given content.

Acknowledgments. This paper has been supported by the CASAM project (www.casam-project.eu).

References

1. Mark, B., M., J.J.: Audio-based event detection for sports video. In: Lecture Notes in Computer Science, Volume 2728/2003. (2003) 61–65
2. Baillie, M., Jose, J.: An audio-based sports video segmentation and event detection algorithm. In: Computer Vision and Pattern Recognition Workshop, 2004 Conference on. (2004) 110–110
3. Huang, R., Hansen, J.: Advances in unsupervised audio segmentation for the broadcast news and ngsu corpora. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). Volume 1. (2004)
4. Panagiotakis, C., Tziritas, G.: A speech/music discriminator based on rms and zero-crossings. **7**(1) (2005) 155–166
5. Scheirer, E., Slaney, M.: Construction and evaluation of a robust multifeature speech/music discriminator. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97. Volume 2. (1997)
6. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. *Speech and Audio Processing, IEEE Transactions on* **10**(5) (2002) 293–302
7. Hyung-Gook, K., Nicolas, M., Sikora, T.: MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval. John Wiley & Sons (2005)
8. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Fourth Edition. Academic Press, Inc. (2008)
9. Misra, H., et al.: Spectral entropy based feature for robust asr. In: ICASSP, Montreal, Canada, 2004. (2004)
10. Pikrakis, A., Giannakopoulos, T., Theodoridis, S.: A computationally efficient speech/music discriminator for radio recordings. In: 2006 International Conference on Music Information Retrieval and Related Activities (ISMIR06)
11. Pikrakis, A., Giannakopoulos, T., Theodoridis, S.: A speech/music discriminator of radio recordings based on dynamic programming and bayesian networks. *Multimedia, IEEE Transactions on* **10**(5) (2008) 846–857