FALA 2010
VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop

Using Fingerprinting to Aid Audio Segmentation

Carla Lopes[1,2], *Arlindo Veiga*[1,2], *Fernando Perdigão*[1,2]

[1] Department of Electrical and Computer Engineering, University of Coimbra
[2] Instituto de Telecomunicações – Pólo de Coimbra

{calopes, aveiga, fp}@co.it.pt

Abstract

Audio segmentation is an important preliminary task in audio description systems. In this paper a three-phase audio segmentation scheme is proposed, where the first phase detects silence events, the second phase detects audio repetitions trough fingerprinting and the last phase uses a hybrid HMM/ANN system to classify the remaining segments. Fingerprinting is an important aspect of audio segmentation in broadcast audio, due to the omnipresence of advertisements, jingles and even repeated programs. Detecting repetitions can be extremely fast and accurate and also permits to enhance the consistence of the annotations. Results using the Catalan broadcast 3/24 TV channel are reported.

Index Terms: audio segmentation, fingerprint, jingle detection

1. Introduction

As the amount of multimedia published data grows, the problem of managing the information contained in this data becomes more and more difficult. Tasks like: finding a roar of laughter to reuse it when editing our own audio or video; counting the number of times that a publicity spot occur or transcript broadcast news are practically impossible. Indexing and content-based retrieval are then crucial to handle the large amounts of audio and multimedia data that is becoming available on the web. Manual annotation is useful in some applications and can provide accurate description of the content. An example of this is when an upload is made in YouTube.com and the user insert keywords to describe the content. Nevertheless, in the main situations manual indexing is extremely time consuming, subjective, tedious and expensive. In these cases audio segmentation and classification plays an important role.

The purpose of audio segmentation is to divide an audio clip into several segments so that each segment contains only one class of audio. Audio signals which include speech, music and environmental sounds are important types of media. The problem of distinguishing audio signals into these different audio types is thus becoming increasingly significant. Although there are many approaches to audio segmentation they are focused on a narrow type of audio such as speech/music separation, speaker recognition and music structure extraction.

In this paper we propose to index broadcast news audio documents in five broad-classes: speech, music, speech with music in background, speech with noise in background and other. After a signal processing section, audio indexing is made up of two main sections: a segmentation section and a classification section. The first one uses silence information and fingerprint information to segment the original audio signal while the second performs a classification in term of five classed within each unlabeled segment. The classification is made by means of a hybrid ANN/HMM system.

2. Hierarchical Audio Segmentation

Broadcast news data are usually stored in long files almost impossible to analyze as a whole. When Viterbi decoding is involved, as in the present work, attention must be paid to the length of the sequence to decode, because the decoding tree grows in such a way that becomes impossible to handle. In view of this we propose a two-phase audio segmentation where the first phase detects silence events in the audio signal in order to limit the length of the segments to classify. We also use make a fingerprint of all the audio session in order to find repetitions in it. The next subsections describe the both detectors.

2.1. Silence Detector

The silence detector is a simple one, based only on the energy of the signal based on a window of 200ms with a shift of 100ms. Energy is computed in dB and when there are segments larger than 1 second below a threshold a silence event is annotated.

2.2. Repetitions detector

Audio fingerprinting refers to a condensed representation of an audio signal that can be used to identify an audio sample or quickly locate similar items in audio streams. We use a fingerprinting system where a 32-bit binary pattern is computed for each frame of about 200ms. The frame rate is 50 frames per second, allowing enough time resolution. The signal is first down sampled to 8 kHz and a spectral analysis is performed with a mel filterbank with 33 channels. The resulting spectrogram is binarized into 32 bits per frame, with a 1-bit, essentially, when there is a spectral peak [4].

The searching method is very simple. It corresponds to count the matching bits between the signature and audio binary patterns, in each frame, when the signature pattern slides over the audio pattern, in order to compute the mean bit error rate. When the bit error rate decreases below a threshold, a match is encountered. Modern processors have a special instruction to count bits, which turns this technique even faster.

In the present case we have used every reference label of the training database as a possible pattern that may repeat. We found thousands of repetitions with this method. However, as the labels of the repeated segments are not always the same, we implemented a voting method to give to that segment the label most referenced.

The information of the two detectors (silence and repetitions) are combined and the output is the original audio signal with silence and music events annotated. In Figure 1 it is showed an example where the upper sequence correspond to the reference sequence annotated in terms of the five classes and the bottom sequence the output of the hierarchical audio segmentation. The grey segments remain unclassified and each one of these segments will be given to an audio classification system in order to annotate the entire audio file. Section 3 describes the audio classification system.

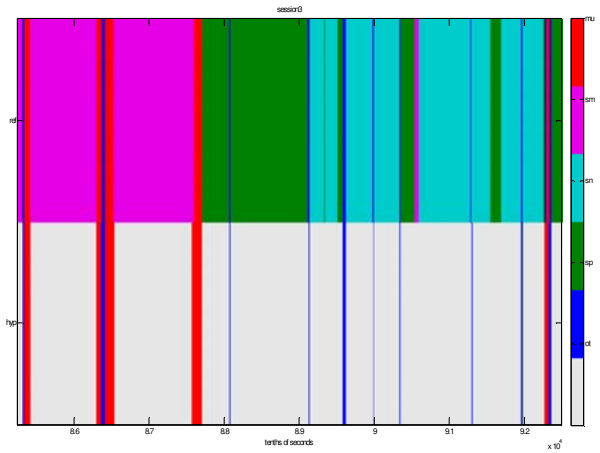

Figure 1: *Hierarchical audio segmentation example.The color coding is the following: mu - music:red; sm – speech with music: magenta; sn – speech with noise: cyan; ot – other: blue.*

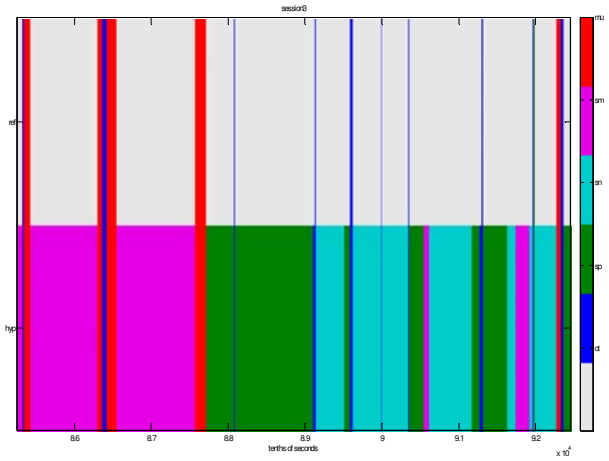

Figure 2: *Audio Classification.*

3. Audio Classification

Audio classification refers to not only the attribution of the correct sequence of labels but also to the labels' boundaries. The audio classification is based on a Viterbi decoding which is applied to each unlabeled segment (grey segments of Figure 1 and 2). The used approach uses a hybrid MLP/HMM system.

3.1. Hybrid MLP/HMM

An MLP network consisting of an input layer (with 200 hidden nodes) and an output layer with five nodes, one for each classe to classify. The 48 parameters described in Section 4.1 were used as standard input features, and a context window of 10 frames in the left and in the right of each frame was considered in the input layer. The softmax function was used as the activation function of the output layer, so that the output values are interpreted as a posterior probability of each class. All the weights and bias of the network are adjusted using batch training with a resilient back-propagation (RP) algorithm [1], so as to minimize the minimum-cross-entropy error between network output and the target values.

In the proposed hybrid approach we considered that the output predictions of the MLP correspond to class posterior probabilities for the input features and we use them as local probabilities in HMM. HMM acoustic models were built for all classes by using HTK 3.4 **Erro! A origem da referência não foi encontrada.**. Each event was modeled by a 10-state left-to-right HMM and each state shared the same MLP output. The HMM also shares the transition probabilities, which were adjusted in order to the model have the same mean duration as the corresponding event. We used HTK with some changes in order to replace the usual Gaussian mixture models by the normalized MLP outputs values and class priors.

In Figure 2 shows an example where the upper sequence corresponds to the output of the hierarchical audio segmentation system and the bottoms sequence the output of the audio classifier. Comparing this figure with Figure 1 (with the reference sequence in the upper part), we can see that most of the events were correctly recognized.

4. Experiments

The Catalan broadcast news database from the 3/24 TV channel was used for both training and testing the segmentation and classification systems, in the context of the Albayzin 2010 Evaluation Campaign [1]. It includes recorded PCM audio at 16-bit resolution and 16 kHz sampling frequency. The training set consists of 16 audio files and the test set of 8 audio files. The training material is labeled in terms of five classes:

1. ***Speech*** [sp].
   Clean speech in studio from a close microphone.
2. ***Music*** [mu].
   Music is understood in a general sense.
3. ***Speech with music*** in background.
   Overlapping between speech and music classes.
4. ***Speech with noise*** [sn] in background.
   Speech which is not recorded in studio conditions, or it is overlapped with some type of noise (applause, traffic noise, etc.), or includes several simultaneous voices (for instance, synchronous translation).
5. ***Other***.
   This class refers to any type of audio signal (including noises) that doesn't correspond to the other four classes

There is a great irregularity within the classes' distribution as depicted in Figure 3.

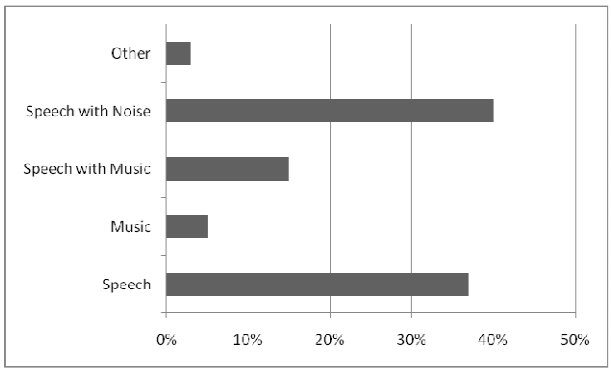

Figure 3: *Distribution of classes within the Catalan broadcast news database.*

The proposed evaluation task consists of segmenting the broadcast news audio files into the referred to five classes.
The performance of the segmentation system is evaluated by means of the metric proposed by the Albayzin 2010 Evaluation Campaign which is based on the NIST speaker diarization task. The metric computes for each class (except *other*) the amount time incorrectly identified (deletion and insertion errors) with respect to the total duration of the respective class. Similar to other metrics used in speech segmentation like *Agreement*¸ a tolerance in a window of ±1seg is given because manual alignments are prone to subjectivity. Considering *Dur* as duration, the metric may be defined as:

$$Error = = \frac{1}{4}\sum_{i=1}^{4}\frac{Dur\left(deletions_{Class_i}\right)+Dur\left(insertions_{Class_i}\right)}{Dur\left(Class_i\right)}. \quad (1)$$

4.1. Features

Neural Networks are capable of incorporating all kinds of input features and adjust itself in such a way that the optimal combination of these features is found for classification. Exploiting this potentiality, input features derived from two different parameterization algorithms are combined: standard MFCC and an additional set. Since music requires more frequency resolution than speech, we considered 16 cepstral coefficients, including c_0, and not the usual 12 coefficients. Table 1 shows the additional set of features used. The feature vector comprises 48 features (16 cepstral coefficients + additional set plus first order derivative). Speech is analyzed every 100ms with a 200ms Hamming window. The classifier gives a sequence of labels, with a minimum duration of one second. The decoder is applied to the segments between the pre-determined silence/music events and not to the entire session.

Table 1. *Acoustic Feature Set used in combination with standard MFCC Features.*

Number	Feature description
1	Frame Energy in dB
2	Zero Crossing Rate
3	Spectral Centroid
4	Spectral roll-off - 90%
5	Max normalized correlation coefficient in each frame (harmonicity measure)
6	Frequency corresponding to the max normalized correlation coefficient
7	Harmonicity measure (proportional to the harmonic duration)
8	Spectral Flux

4.2. Results

A final classification error of 20.68% was obtained in the 16 sessions of the training part of the database. Table 2 shows the results per class. The class music achieved the best performance while speech with noise achieved the worse.
Quite different is the final classification using the audio test material. The final error rate almost doubled! Music kept the best class performance nevertheless it degraded 8% in the error rate. One reason for this is that the total time of repetitions and silence encountered in test sessions is only about 12% compared with the 65% on the reference sessions. In fact, we found 4427 non-overlapping segments in the reference sessions that repeat at least twice in all sessions (that we have called jingles). This corresponds to 65% of the total reference session's time.
The evaluation corresponds to finding the silences and jingles, and to the Viterbi decoding of the segments between silences and/or jingles. The code was mainly implemented in Matlab and the used machine was a Cray CX1.

Table 2. *Classification error results using training material.*

```
=====================================
The error-rate of the class mu : 13.62%
The error-rate of the class sp :  22.78%
The error-rate of the class sm :  20.10%
The error-rate of the class sn :  26.21%
=====================================
The final error-rate : 20.68%
=====================================
```

Table 3. *Classification error results using test material.*

```
=====================================
The error-rate of the class mu :  21.43%
The error-rate of the class sp :  48.03%
The error-rate of the class sm :  51.66%
The error-rate of the class sn :  48.49%
=====================================
The final error-rate : 42.40%
=====================================
```

5. Conclusions

Although the results were lower than we expected, the method of finding repetitions with fingerprinting is important in audio segmentation of broadcast audio where repetitions are always present. The observed differences from the reference and test results can be explained to overtraining but also to the segmentation method, which relies on the repetitions of audio segments. However, the method exploits the inconsistencies that exist in the annotations, which contributes also to the observed errors.

6. Acknowledgments

Carla Lopes would like to thank the Portuguese foundation: Fundação para a Ciência e a Tecnologia for the PhD Grant (SFRH/BD/27966/2006).

7. References

[1] http://fala2010.uvigo.es/index.php?option=com_content&view=article&id=60%3Aaass&catid=36&Itemid=65&lang=pt#_ftn1
[2] Riedmiller, M. and Braun, H., "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in Proc. ICNN, San Francisco, CA, 1993, pp. 586–591.
[3] Young, S. *et al*, The HTK book. Revised for HTK version 3.4, Cambridge University Engineering Department, Cambridge, December 2006.
[4] C. Neves, A. Veiga, L. Sá, and F. Perdigão, "Audio Fingerprinting System for Broadcast Streams," in Proceedings of the Conference on Telecommunications (ConfTele2009), pp. 481-484, 2009.