Audio Content Description of Movies

Virginia Maria Martins Barbosa

June 7, 2011

Abstract

With the ever-growing quantity of multimedia documents uploaded on the Web, automatic description of the audio content of videos can provide very useful information, to index, archive and search multimedia documents. Audio Event Detection (AED), aims to substantially enhance the access to audio in multimedia content. Preliminary experiments with a sound effects corpus showed good results for training models. However, the performance on the real data test set, where there are overlapping audio events and continuous background noise is lower. This work describes the AED framework and methodologies used to build 6 Audio Event detectors, based on statistical machine learning tools (Support Vector Machines). The detectors showed some improvements achieved by adding background noises to the training data, comprised of clean sound effects that are quite different from the real audio events in real life videos and movies. A graphical interface prototype is also presented, that allows browsing a movie by its content and provides an audio event description with time codes.

Keywords

Audio Event Detection, audio content analysis, semantic video search

1 Introduction

More and more multimedia content is uploaded on the Internet. Take for instance video sharing websites like YouTube, they make it possible for anyone with an Internet connection to upload a video that could be watched worldwide within a few minutes. In fact, every minute, 35 hours of video is uploaded to YouTube\(^1\). Consequently, it has become increasingly difficult to make a simple content search. The grand majority of video search engines are based on textual tags inserted by the users. Audio content retrieval could improve greatly the semantic video search. Yet, there are other application areas. With audio content retrieval it is possible to obtain an audio description of a movie. This description provides time segments with music and all kinds of sounds (e.g. gun shots, explosions, singing birds, cars passing-by, etc). This information can be very helpful to hearing impaired people, enabling them to follow the synopsis better or to be able to browse the movie in terms of audio events, through a graphical interface.

\(^1\)http://www.youtube.com/t/press_statistics (last visited in May 2011)
As for blind people, an extended audio content description is of great value, because much more details from the scenes are available in audio form.

Audio Event Detection and Classification experiments reported in this work, concern non-speech audio events only. A sound effects (SE) corpus was used to train the detectors. Sound effects are artificially created sounds, usually in optimal recording conditions, or sounds that are subject to some enhancing process, which make them very clean. The corpus showed good results on validation sets, comprised of sound effects, but this was not the case with the real life data test set [Bugalho et al. 2009]. The real data test set contains movies, documentaries, talk shows and broadcast news shows. The audio in these test files is likely to have background noise, many audio events occur simultaneously and recording conditions are not optimal. This gap between the train/validation sets and real life videos and movies test sets may explain the significant difference in performance.

The main goal of the present work is to overcome the data mismatch, improve the detection of audio events and increase the performance of the classifiers. To accomplish these tasks, the sound effects corpus was enriched with background noise. For six concepts, several audio events detectors were developed and tested, using Support Vector Machines (SVMs), a statistical machine learning tool. It is important to compare the performance of the new detectors with the previous ones, hence the results obtained will be subject of a thorough analysis. It would be interesting to see the results accomplished so far in a practical and user-friendly manner. An audio event viewer, could also be used as a debugging tool. The other goal is to develop a graphical interface prototype, specifically designed to browse motion pictures semantically using its audio track. The interface prototype presented in this work, is an interactive Web application which provides an audio content description of a movie. The most interesting feature of the application is the integration with the AED software. The system provides the possibility to upload new videos and choose which audio events to detect.

2 State of the Art

2.1 L²F Audio Event Detection Framework

The AED system structure is illustrated in Figure 1. The diagram represents the processing chain of a movie, starting with any video or movie. Five major modules comprise the chain.

First the audio signal is retrieved from the video file using the FFmpeg\(^2\) software. In the Speech/Non-speech (SNS) Classifier module, an in-house SNS detector is applied to separate speech and other sounds associated with the human voice such as laughing, crying, screaming, from other events. The segmentation module comes next, where an Acoustic Change Detector (ACD) determines the audio events limits in time. The ACD is used to segment the audio stream into acoustically homogeneous segments. The feature extraction process deals with different types of features, some of which commonly used in ASR, like Mel-Frequency Cepstral Coefficients (MFCC) [Bugalho et al. 2009]. Others are low-level MPEG7 audio descriptors more specific to the problem, such as Zero Crossing Rate (ZCR) and Audio Spectrum

\(^2\)http://www.ffmpeg.org/ (last visited in May 2011)
Envelope (ASE) [Vinet, H et al. 2002]. In the audio event inference module, various machine learning methods can be used to provide a final classification of the audio events. The Audio Event Detection is frame-based, but the ACD module provides time segments where only one Audio Event is supposed to happen. Finally, the information from each frame is combined to generate a final XML annotation file for each concept. Confidence measures are estimated considering the number of frames a segment contains. This file follows the format described in the MPEG7 standard schema [Martínez et al. 2002].

2.2 Audio Event Inference

It is very difficult to do a state of the art in terms of audio events detection. Different techniques are used and in literature there is no clear solution. The reason why a comparison between methods is hard to achieve, bears with the fact that each study approaches the AED problem in its own fashion. Many times they focus on specific cases, have a controlled quality of audio, work with small domains or with a limited set of audio events and background sounds. Past CLEAR evaluations had an AED task with a focus on event detection during meetings [Temko et al. 2007]. The European CUIDADO project concerns musical instruments [Vinet, H et al. 2002]. Other studies deal with sports games [Huang, Q. and Cox, S. 2010]. One study focused on AED in soccer videos, which have a very structured domain; cheering sounds vary during a game and its volume is often correlated with particular events such as when a team scores or when there are fouls [Xu et al. 2003]. The SOLAR project is a system that focuses on a problem similar to ours, but it presented results for a limited set of thirteen concepts [D. Hoiem et al. 2005]. Each study has its own data set, each extracts different features thus a comparison of methods is difficult. Even though this thesis focus on a limited set of audio events (six semantic concepts), it deals with a more complex domain, since these concepts are to be detected in any kind of videos, from movies to user-generated content. In these videos, audio events usually occur simultaneously, continuous background noise often blurs the audio events of interest, music and speech overlapping, etc. To model audio events generic enough to accomplish this goal is a very challenging task. The machine learning methods considered in this work are low-level classifiers. There two major groups: supervised and unsupervised learning. Its goal is to predict a class label after having seen a number of examples from the training set. Oppositely,
unsupervised learning is given only unlabeled examples. A supervised learning approach will be adopted in this work: Support Vector Machines for the good prediction capabilities the classifiers demonstrated in the early experiments conducted in the beginning of the VIDIVIDEO project.

3 Baseline Audio Event Detection System

A sound effects corpus was used to create a baseline. Despite the viability of using a sound effects corpus to train the detectors, the final goal is to use the audio event detectors in a real life scenario. Only a small set of real life movies and documentaries was manually annotated, in order to measure the detectors performance. This manual annotation was performed for only six concepts: **Birds, Crowd Applause, Dog Barking, Sirens, Telephone Ringing Bell** (old telephone, not digital) and **Water**.

The evaluation corpus consists of 13 video files with a total duration of about 13 hours. The set gathers movies, documentaries, talk-shows and broadcast news. These movies were manually labeled in terms of the six audio events. A set of 237 features were extracted from the audio files. Each feature is retrieved from a 40 ms analysis window with a sliding window of 20 ms. The mean value and standard deviation is added to each feature using 25 frames to each side, to provide temporal context. A Principal Component Analysis (PCA) step is introduced, where the most discriminative features are selected, unimportant and redundant data is removed. The final set is reduced to 150 features. The same feature extraction is performed for the train, validation and test sets. In this work one used the strategy of one-against-all binary SVM classifiers. For each target audio semantic concept a classifier is built, one that distinguishes the event from the other events, grouped in a so-called “World model”.

The performance of the classifiers is measured in terms of the F-measure, a measure that combines Precision and Recall. The closer to 1 the better.

\[
F\text{- measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]  

The baseline performance of the classifiers in the SE corpus achieved an average F-measure of 0.93, which is a very good performance. The real life test set achieved an average F-measure of 0.31. This work attempts to leverage the data mismatch between the validation and the test sets, with the addition of background noise to the sound effects corpus.

4 Experimental Results

A group of sound effects concepts was selected to be added as background noise. Two signal-to-noise ratios (SNR) were tested. SNR compares the level of a desired signal, in this particular case the audio events, to the level of background noise. The higher the ratio, the less obtrusive the background noise is. The clean sound effects corpus was enriched with SNR25 and SNR10.
4.1 Experiments with the Validation SE Corpus

The sound effects corpus with background noise (BN) was used to train one-against-all detectors for each one of the six concepts. The models obtained an average F-measure of 0.89 in the validation set. There was a slight decrease in performance, when compared to the validation results for the clean SE detectors, which achieved an average F-measure of 0.93. This decrease was expected, because the mixed SE audio files have much more spectral content, and the signal features are not as discriminative as in the clean SE. Despite the performance slight decrease, better results are expected in the test set. The background noise was introduced to smooth the difference between SE and real life videos, where audio events are usually corrupted by background noise, so these audio files also have more spectral content than clean SE.

4.2 Experiments with the Evaluation Corpus

The real life test set comprised of movies and documentaries, was used to test the BN detectors.

Table 1 presents the results for the clean detectors, and the new BN detectors, for each one of the files in the test set.

Table 1: Results for BN and clean detectors in terms of Precision, Recall and F-measure.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Test file</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Clean</td>
<td>Noise</td>
<td>Clean</td>
</tr>
<tr>
<td>Castelli</td>
<td></td>
<td>0.15</td>
<td>0.09</td>
<td>1.00</td>
</tr>
<tr>
<td>Birds</td>
<td>Kosovosodo</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Populonia</td>
<td>0.55</td>
<td>0.44</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.29</td>
<td>0.18</td>
<td>0.36</td>
</tr>
<tr>
<td>Crowd</td>
<td>Portugal-Coracao</td>
<td>0.93</td>
<td>0.96</td>
<td>0.23</td>
</tr>
<tr>
<td>Applause</td>
<td>Preco-Certo</td>
<td>0.74</td>
<td>0.92</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.78</td>
<td>0.94</td>
<td>0.31</td>
</tr>
<tr>
<td>Dog</td>
<td>Corazon Batida</td>
<td>0.88</td>
<td>0.86</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>Monterias</td>
<td>0.69</td>
<td>0.67</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.76</td>
<td>0.75</td>
<td>0.12</td>
</tr>
<tr>
<td>007</td>
<td></td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
</tr>
<tr>
<td>Sirens</td>
<td>Die Hard 4</td>
<td>0.00</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Telejornal</td>
<td>0.65</td>
<td>0.63</td>
<td>0.37</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.23</td>
<td>0.18</td>
<td>0.24</td>
</tr>
<tr>
<td>Telephone</td>
<td>The Aviator</td>
<td>0.32</td>
<td>0.39</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>Telephone</td>
<td>0.27</td>
<td>0.32</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.28</td>
<td>0.34</td>
<td>0.82</td>
</tr>
<tr>
<td>Water</td>
<td>Paesaggio</td>
<td>0.09</td>
<td>0.09</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>0.39</td>
<td>0.41</td>
<td>0.25</td>
</tr>
</tbody>
</table>
In average, results show a slight improvement when background noise is added to the training set. For the Dog Barking audio event, for example, adding background noise led to gain in F-measure of about a factor two. Hunting videos were used to detect the Dog Barking concept. These videos have more background noise content (wind and other animal sounds, step sounds, people talking, gun shots), which may explain the significant gain for this particular audio event. In the CastelliDellaLunigiana (Birds) and PaesaggioRitrovato (Water) videos, even though almost all positive frames were correctly classified, many negative frames were misclassified as positive, hence the high Recall score and low Precision. This situation occurs in both the clean and noise detectors. The CastelliDellaLunigiana and Populonia documentaries (Birds) barely have background noise content. Although the Birds sounds appear in outdoors scenes, surrounding environment is silent. Sirens and Crowd Applause are very loud sounds, and the addition of background noise could have mislead the detectors making the features less discriminative. For a given concept, some detectors showed good or low performance depending on the movie. This could be explained by the diversity of the sounds in these movies. The results suggest that the performance of the detectors is influenced by the type of sounds in the test set, whether they are similar to the training examples or not. There are cases where the background noise added to the sound effects does not match the sounds in the test files. For the Siren concept, 007 - A View to a Kill and Die Hard 4 movies have overlapping audio events, the sound of crashing cars, helicopters is often present in the background during the sirens events. Overlapping audio events were not accounted for in the training set. In movies, audio events are likely to occur with music in the background; the so called film score, used to create ambiance. This kind of music was not added as background noise.

From these results it is important to highlight the fact that the amount and diversity of data necessary for training is crucial. It is difficult to achieve a good model generalization even with the addition of several different background noise concepts. There are many factors that can influence the classification and, simulating them all in a training set is a very difficult task. Nevertheless, it is possible to conclude that some detectors became more robust with the addition of noise.

### 4.3 Additional Experiments with YouTube Videos

YouTube videos are mostly recorded by amateurs and thus contain more noisy information. Unlike the movies, documentaries and talk-shows which have sound post-processing, most of YouTube videos were recorded in poor audio conditions, generally with background noise made by the wind or the camera movement. YouTube videos were retrieved to illustrate the six concepts. These videos were not manually annotated, just globally labeled as being instances of the audio events for which they were collected. The target audio event is mostly represented, but the files contain a mixture of audio (background sounds, some voices). For each concept 6 to 9 videos were retrieved.

Since each file contains mostly the audio event for which it was retrieved, and there is no label marking the beginning and the end of the audio events, a file classification evaluation is more appropriate. Each file is assigned the concept which obtained the highest confidence. The classification accuracy was measured
using the following metric:

\[
Acc = \frac{\text{Number of Correctly Classified Files}}{\text{Total Number of Files}}
\]  

Birds detector is not discriminative, since it achieved considerably high confidences in almost every audio event. To reduce this bias on the detection of the other audio events, a score normalization was performed. Table 2, shows the file classification results after normalization. The values in the diagonal of the matrix, in bold, represent the number of correctly classified files. The values that are not in bold, represent the number of misclassified files, for each detector.

Table 2: File classification results in the YouTube videos, after normalization.

<table>
<thead>
<tr>
<th>Files</th>
<th>Birds</th>
<th>Crowd Applause</th>
<th>Dog Barking</th>
<th>Sirens</th>
<th>Telephone</th>
<th>Water</th>
<th>Acc (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Birds</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>Crowd Applause</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>Dog Barking</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>Sirens</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>Telephone</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Water</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>60</td>
</tr>
</tbody>
</table>

Even though some detectors still show some confusion with different audio events, they correctly classify more files of target concept. The Birds detector may be a less discriminative detector, due to the addition of background noise. This misleads the detectors to consider features that are not representative of the concept. The Crowd Applause detector obtained the lowest performance, which may be explained by the poor quality of some Crowd Applause YouTube videos. Some of these videos show a large crowd of people cheering and screaming while clapping. The high frequencies of these sounds can explain the confusion with the Birds audio event. Sirens and Telephone achieved a high performance. This may be explained by the fact that the sounds in these files are very similar to the ones used to train the detectors.

The diversity in the data sets is the key to the AED problem. However, it is quite hard to collect, storage and process the amount of data necessary to achieve high detection performances.

5 Graphical Interface Prototype

The graphical interface prototype contains three main functionalities: Visualization Mode, Debug Mode and Upload Mode.

The interface provides an audio content description of a movie. Figure 2 shows a screen capture of one the views of the interface. When choosing a specific event, a list of time codes is available. By selecting one of the time code buttons on the right side of the video, the user can move forward or backward through the different segments of the film, according to the classification. The debug mode has the possibility to compare classifications of two detectors, with and without background noise, and also
with the manually annotated reference. The most interesting capability of this system is the possibility to submit new multimedia content and select which audio events to detect. The AED software is integrated with the graphical interface, thus allowing the user to run the software “on the spot”, and promptly browse the video uploaded in terms of audio events.

Figure 2: One of the views of the Graphical Interface Prototype.

5.1 General Architecture

Figure 3 illustrates the architecture of the web application.

Figure 3: Architecture of the Web Application.

- **Web Browser:** Users interact with the system through a Web Browser. The web page is coded in Hypertext Markup Language (HTML). The dynamic content is generated using JavaScript with the
jQuery\(^3\) library. The video player used to embed video streams into the web page is Flowplayer\(^4\).

- **Web Server**: The Web server is responsible for fetching information from the database and sending it to the Browser. It also acts as the execution server’s client. Upon the submission of a video, a connection is established. It is implemented in php.

- **Execution Server**: The Perl execution server runs continuously, passively awaiting incoming connections. When a connection is established, the server starts the AED software on the video uploaded. The big challenge was to understand how could the AED software be integrated in the graphical interface prototype. The AED software are runs in Linux machines, and needs access to in-house software, and numerous tools. It is mostly shell scripting modules. The solution of an execution server seemed to be the most practical. The Perl server and the AED software run on an internal L\(^2\)F machine with AFS (Andrew File System) access. At the end of the execution of the AED software, the results of the detection are introduced in the database, and the execution server returns a message to the Web Server notifying that the AED finished.

- **Database**: In terms of data storage, a mechanism was devised to enable the information trading between the execution server, which produces content, and the web server which consumes that content and sends it to the Web Browser. The objective was to have a functional version of the interface and, since both components have access to the AFS, a flat file database seemed to be the fastest solution to implement an information sharing mechanism. Although, in the future a relational database could be used.

## 6 Conclusion & Future Work

With the ever-growing quantity of multimedia documents uploaded on the Web, automatic description of the audio content of videos can provide very useful information, to index, archive and search multimedia documents. In literature one finds different approaches to the AED problem, there is no clear solution. Support Vector Machines classifiers were adopted in this work for the good prediction capabilities showed in preliminary experiments carried out by the L\(^2\)F team. A SE corpus was used to train the SVMs classifiers. The AED experiments conducted with the SVMs classifiers show that the results for the clean SE data set, are much better than for real life videos, where recording conditions are different, multiple audio events may be combined and background noise is often present. This work attempted to minimize the data mismatch between the sound effects data and the real life data, by adding background noise to the SEs, in order to simulate real life audio in videos. In average, results show a slight improvement when background noise is added to the training set. For a given concept, some detectors showed good or low performance depending on the movie. This could be explained by the diversity of the sounds in these movies. The gap between sound effects data and real life data was

\(^3\)http://jquery.com/ (last visited in May 2011)

\(^4\)http://flowplayer.org/ (last visited in May 2011)
somehow reduced. Thus, the advantage of using implicitly labeled corpora still holds, sparing the time-consuming task of manually labeling, but with the addition of background noise some classifiers became more robust. Additional experiments with a YouTube test set were also presented. As more and more user-generated content is available on the Web, AED could provide content-based tag suggestion, as an alternative to textual tags inserted by users. The second part of the work focused on the development of a graphical interface prototype. A Web application that allows users to submit and search videos in terms of automatically detected audio events. The main feature is the ability to upload videos and execute the AED software “on the spot”. Hence, the SVMs detectors developed during this thesis, were integrated in this interactive application.

As future work, it would be interesting to further investigate the use of visual cues to improve the detection of the audio events. Experiments with Gaussian Mixture Models could be interesting as this is a common technique in other tasks such as speaker and language identification. Regarding the graphical interface, new functionalities could be added. The possibility to add labels to the videos and identifying the occurrence of audio events. If properly implemented, this feature could evolve to gathering enough information to create a training set with real life data. For example, using Crowdsourcing games or competitions, for collaborative tagging.

References


Vinet, H et al. (2002). The cuidado project: New applications based on audio and music content description. In In Proceedings of the ICMC.


Crowdsourcing is the act of outsourcing tasks, traditionally performed by an employee or contractor, to an undefined, large group of people or community (a “crowd”), through an open call.