HIERARCHICAL LANGUAGE MODELING FOR AUDIO EVENTS DETECTION IN A SPORTS GAME

Qiang Huang and Stephen Cox

School of Computing Sciences, University of East Anglia
Norwich, UK.
{h.qiang, s.j.cox}@uea.ac.uk

ABSTRACT

We investigate the automatic labelling of "events" from an audio recording of a sports game. We describe a technique that utilises a hierarchy of language models, which are a low-level model of acoustic observations and a high-level model of audio events that occur during a game: these models are integrated using a maximum entropy approach. Our models of the audio events also utilise duration and voicing information as well as spectral content, and we show that further discrimination between events is possible using these features. Results on different tennis games show that the use of these techniques is better than using an approach that does not use modelling of dependencies between frames and events or extra information in the form of duration and voicing.

***Index Terms*—** Language Modeling, Audio Event Detection

1. INTRODUCTION

The long-term goal of the research reported here is to develop systems that are capable of understanding, and thus participating in, complex human transactions. At this early stage of the project, we need to develop tools for classification of the video and audio "events", and here, we address the problem of identifying the class of a certain audio event in a tennis game.

There has been recent interest in applying multimodal analysis techniques to identity automatically events occurring within sporting games, describe their contents, explore their dependencies, and summarize logical relations among them. The approach is to utilize both video and audio signals to attempt to identify significant events. Visual features are clearly a highly important source of information about events and interactions [1, 2, 3, 4]. But some interesting results in [1] show that using only visual features does not yield very high performance in event recognition, and this has shifted the focus towards incorporating audio information. The use of audio information has some advantages in efficiently and effectively detecting events in the domain of sports video, such as the tennis match video explored in this paper. The task of identifying such events is rather different from that of speech recognition, where the "events" are words or phones and occur sequentially. This is because events in sports games can occur simultaneously, not all events are of interest or importance, and events can have very different durations (e.g. the striking of a ball can be a significant event, as can a long ovation from a crowd).

In a tennis match, there are some characteristic audio events that include ball striking sounds, crowd roars, commentators' speech, the chair umpire's speech, line judges' and players' shouts etc. These can all be used in different ways to infer the state and progress of the game, and when combined with the events detected by a computer vision system, are a powerful source of information. For example,the voice of the chair umpire furnishes us with information about the scores and the long-term progress of the match, whether there is a challenge etc. and the applause, gasps, cheers, roars etc. of the crowd can naturally be used as an indication of the start or the end of a point. These audio events provide complementary information, which is overlapping, and which needs to be gathered at different time-scales.

In this paper, we present a hierarchical framework to detect audio events in live tennis matches. The fundamental idea is that we convert the audio event detection task into the problem of optimizing language models in a two-level hierarchical structure. At the low level, a language model is trained over the output symbol sequence obtained from the observed acoustic features, whilst at the high level, an audio-event based language model is trained. The link between the two levels is the mapping from the low-level features to high-level audio events. The construction of the language models at two levels and the link between them are optimized using maximum entropy (ME).

The rest of this paper is organised as follows. Section 2 reviews related work. Section 3 explains the framework and theory of this hierarchical language modelling technique. Section 4 describes the data used, and experiments and evaluation are presented in Section 5. We end with conclusions in Section 6. Note that in this paper, the term "language model" is used by analogy with language models in speech recognition to describe a probabilistic model of sequences of frames, and a probabilistic model of audio events.

2. RELATED WORK

Event detection in sports games and the highly similar task of automatic segmentation of meetings have recently become important research areas. Some approaches attempt to construct a general framework, while others focus on specific sequence labelling tasks. The former usually utilize machine learning algorithms [5, 6, 2], such as hidden Markov models (HMM) [1], support vector machines (SVM) [5], conditional random fields [5, 6] and focus on optimization of model parameters. The latter methods pay more attention to specific labelling tasks, such as audio sequence labelling and video segmentation [7, 1, 4, 2]. In these methods, lower-level audio and visual features are often separately or jointly used to detect the audio events or segment videos, and some good results have been obtained.

Language modelling has, of course, been crucial in the development of speech recognition systems, but has not been utilised much in audio event detection. The work presented here focuses on combining low-level and high-level event modelling in a hierarchical framework that takes into account the dependencies between the two

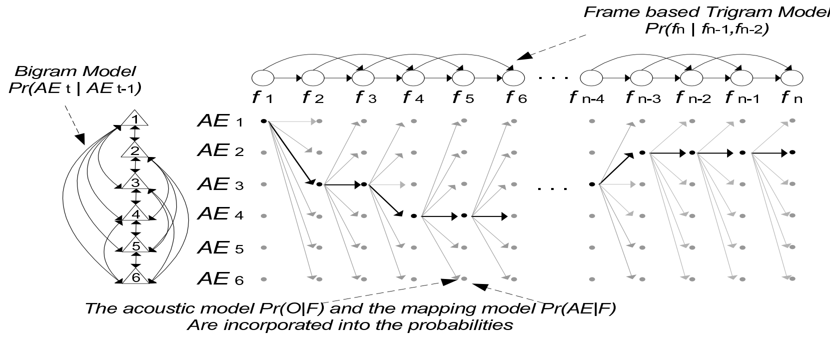


Fig. 1. Viterbi decoding algorithm

levels. The theoretical framework will be described in detail in the next section.

3. THEORETICAL FRAMEWORK

In this section, we introduce the hierarchical framework and show how the different elements within it are estimated. We then describe the application of maximum entropy (ME) to the density estimates of the observed audio features and show how the estimates from ME are integrated these information into our framework. We also describe the use of duration models and pitch in modelling the acoustic events in a game of tennis.

Our goal is to classify a sequence of acoustic features O as a sequence of *audio events, AE*. In a maximum likelihood framework, the most likely sequence AE^* is obtained as

$$AE^* = \arg\max_{AE} \Pr(AE|O) \quad (1)$$

In the usual way, using Bayes' theorem:

$$AE^* = \arg\max_{AE} \Pr(O|AE)\Pr(AE) \quad (2)$$

We now introduce an extra "latent" variable F, so that we can re-write equation 2 as

$$\begin{aligned} AE^* &= \arg\max_{AE} \sum_F \Pr(O|F)\Pr(F|AE)\Pr(AE) && (3) \\ &= \arg\max_{AE} \sum_F \Pr(O|F)\Pr(AE|F)\Pr(F) && (4) \end{aligned}$$

In equation 4, F represents a sequence of audio event labels, labelling the frames that comprise an example, and $\sum_F$ is read as "sum over all possible label sequences". A label for a frame has the value $\{1, 2, \ldots N_{AE}\}$, where N_{AE} is the number of distinct audio event classes: the label is the most likely audio event associated with the frame, and is estimated from a Gaussian mixture model (GMM) of each audio event.

The three terms in equation 4 can be computed as follows:
1. The term $\Pr(O|F)$ is computed from acoustic models of the audio events: we used GMMs, which are trained using manually labelled data. We assume independence of frames: this patently false assumption is corrected during the later stages of processing. Hence

$$\Pr(O|F) = \prod_t \Pr(O_t|F_t). \quad (5)$$

2. The term $\Pr(AE|F)$ can be modelled as depending on the history of the audio events (approximated here by a bigram) and the probability of an event given a certain frame labelling:

$$\begin{aligned} \Pr(AE|F) &\simeq \prod_t \Pr(AE_t|AE_{t-1})\Pr(AE_t|F_t) && (6) \\ \text{where} \quad \Pr(AE_t|F) &\simeq \Pr(AE_t|f_t, f_{t-1}f_{t-2}). && (7) \end{aligned}$$

Here, AE_t denotes the audio event AE that occurs at time t. $\Pr(AE_t|AE_{t-1})$ corresponds to a bigram "language model" of audio events, which is estimated from the labelled training data. Estimation of the term $\Pr(AE_t|F) = \Pr(AE_t|f_t, f_{t-1}f_{t-2})$ was performed using standard linear interpolation techniques, and the estimates were then smoothed using Maximum Entropy techniques.
3. The probability of the sequence of labels F can be estimated as if it is a sequence of words or phones using a tri-gram model:

$$\Pr(F) = \prod_t \Pr(f_t|f_{t-1}f_{t-2}). \quad (8)$$

Practically, it is not possible to use a model of frame events that is derived from the manual labelling of the frames. In such a model, $\Pr(AE_t = AE_i|AE_{t-1} = AE_i) \simeq 1$, because an event lasts for many frames and all the frames within an event have the same label. We therefore learn a model that is based on the labelling of the training-set frames by the acoustic models. Although this model is errorful, it is a valuable source of information, as will be seen in section 5.

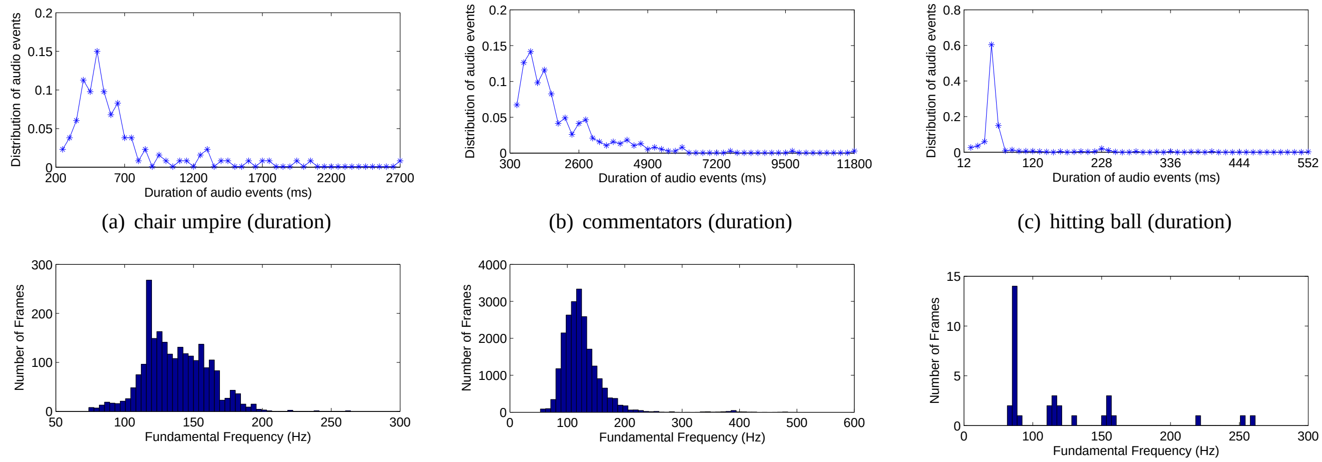
We assume that equation 4 can be approximated by the most likely sequence over all F (as is standard in ASR), in which case we can re-write it as:

$$\begin{aligned} AE^* \approx\ & \arg\max_{AE}\{\Pr(AE_t|AE_{t-1}) && (9) \\ & * \max_F\{\Pr(O|F)\Pr(AE_t|F)\Pr(F)\}\} \end{aligned}$$

Although equation 9 looks complex, the algorithm that solves it is actually very similar to that for connected word recognition from a noisy phone sequence using the Viterbi algorithm [7]. Figure 1 illustrates this. The labels $f_1, f_2 \ldots f_N$ correspond to a sequence of phone labels that have been provided by e.g. a phone loop recogniser. Audio events correspond to words, so that $\Pr(AE_t|AE_{t-1})$ is equivalent to a bigram word model. $\Pr(AE_t|F)$ corresponds to the probability of a word given a phone sequence, and $\Pr(F)$ to a tri-gram model of the noisy phone labels.

We can also make use of specific acoustic properties of the audio events, in this case, pitch within an event and duration of the event.

Figure 2 shows the duration and pitch distribution of three audio events: "chair umpire", "commentator", and "ball hit". The top row



(a) chair umpire (duration) (b) commentators (duration) (c) hitting ball (duration)

(d) chair umpire (pitch) (e) commentators (pitch) (f) hitting ball (pitch)

Fig. 2. Duration and Pitch distributions of three audio events

shows that the duration distributions of the three audio events are quite different: the duration of umpire's voice ranges from 280ms to 750ms, while most of the commentator's segments last for more than 700ms. The impulsive sound of a racquet striking a ball has a mean duration of only about 90ms. Pitch information is a good way of distinguishing between speech and non-speech events. If a pitch estimation algorithm is run on the audio events, the umpire's voice and commentators' voices show that voicing is often detected, and the distributions are similar, whereas the "ball hit" histogram shows very little voicing is detected, although there are a small number of voiced frames caused by the players grunting!

To integrate this information, we first set empirically derived minimum and maximum thresholds of duration and pitch for each audio event. During traceback in Viterbi decoding, the duration and the distribution of each detected audio event is noted. If the label of the decoded audio event is outside its permitted limits set by the thresholds mentioned above, it is changed to the next best event match in decoding, and this process is continued until an event that does not fall outside the bounds of its threshold is found. This is an *ad hoc* approach that we intend to improve and develop later.

4. DATA

We performed our experiments on an audio corpus which consists of four audio tracks, each lasting about 22 minutes, taken from video recordings of two different tennis games. Three of the tracks are taken from the same tennis match but have some variations in audio characteristics. The first track was judged to have fewer overlapping/simultaneous audio events and was selected as a training set ($Training$). Tracks two and three are used as test sets ($Test1$, $Test2$): these have more overlap of crowd noise and speech. The data from the second match forms a third test set ($Test3$).

Each audio track was manually segmented and each segment was labelled with one of six different audio events. These events were:

1. silence;
2. speech from chair umpire;
3. speech from commentator(s);
4. cry from line judge(s);
5. sound of racquet hitting ball;
6. crowd noise.

Although simultaneous events will be of importance later on in our work, for present purposes, any segment of an audio track had a single label applied to it, which was what was judged to be the most prominent event during that segment.

Audio analysis was standard: the audio sequence was windowed into 30ms-length frames with 20ms overlapping from which 26-D MFCC vectors were generated, which consisted of 12-D MFCC coefficients, overall energy, and their first differences. Cepstral mean normalization was applied at the track level.

After the tracks had been manually labelled, each frame effectively had an associated label that is one of the six audio event categories above. We use frame error rate (FER) as our performance measurement throughout these experiments.

5. EXPERIMENTS AND EVALUATION

The order of our experiments was as follows:

1. GMM labelling of the frames only;
2. as above, but with application of the frame based tri-gram language model;
3. as above, but with application of the frame/event mapping model and the event-based tri-gram language model;
4. as above, but with application of the duration and pitch modelling.

Preliminary experiments indicated that a 16 mixture component GMM was appropriate for modelling the audio events of "chair umpire" and "commentator's speech", whereas the other audio events, which are acoustically much simpler, could be well-modelled using only three components. These values could, of course, be exhaustively optimised, but in this work, we focus on the integration of the language models. Table 1 shows the frame error rate over the training- and test- sets. Row 1 shows the error-rates obtained when labelling randomly, using the priors to choose the labels: we include this as a baseline, since the priors are very different across the six classes. Using the GMMs on the training-set, the error-rate is reasonably low, and most of the mis-classification is between the umpire's and commentator's speech. Error-rates are much higher on the

FER	Training	Test1	Test2	Test3
Random (baseline)	53.6%	54.5%	54.9%	45.9%
GMM	18.63%	30.49%	37.34%	44.68%

Table 1. Frame error rate using GMM acoustic models only

test-set, especially the third set, which is from a different match that was (presumably) recorded in a slightly different way. In Table 2, the results of using the frame based tri-gram language model (F-3LM) are listed. We iteratively run this step by using the decoded frame sequence from the previous decoding as the input for the next iteration. Performance here is substantially better on both training and test-set than using only GMMs. The iteration of the decoding gives a small improvement in performance.

#Iteration	1	2	3	4	5
Training	8.81%	8.69%	8.58%	8.62%	8.70%
Test1	17.68%	17.58%	17.20%	17.16%	17.20%
Test2	24.06%	23.90%	23.70%	23.54%	23.41%
Test3	32.19%	32.14%	32.00%	31.95%	31.93%

Table 2. Frame error rate using GMM+Viterbi+F-3LM

	Training	Test1	Test2	Test3
GMM+Vit.+F-3LM	8.70%	17.20%	23.41%	31.93%
GMM+Vit.+F-3LM M-LM	8.68%	17.14%	23.23%	32.05%
GMM+Vit.+F-3LM M-LM+E-LM	8.66%	17.11%	23.10%	31.38%
Improvement	+0.46%	+0.53%	+1.32%	+1.72%

Table 3. Comparison of performances using mapping model and event based language model

Table 3 compares the performances starting with the frame based language model (F-3LM, as in Table 2), the mapping language model (M-LM), and the event based language model (E-LM) are added step-by-step. Comparing with the results using GMM+Vit.+F-3LM, the improvements obtained are small. This may be due to a number of reasons. Firstly, the frame based language model has an excellent ability to correct mis-labelled frames from the GMM, and so the baseline performance is already much better than using GMMs alone. Secondly, at the moment, we are using a "grammar factor" of one, i.e. the weights of the frame-based tri-gram model and the event-based language model are equally balanced. It is likely that increasing the weight of the event-based language model will increase performance, but this is still under investigation. Thirdly, the frame-based tri-gram model is trained on the output from the GMM classifier, which is errorful, although its FER is much lower than the FER on the test-set. Applying the the frame-based tri-gram model to test data does improve performance, but the model is inherently incapable of giving very low error-rates.

The final results listed in Table 4 show that very significant further improvements are obtained when the audio event duration and pitch distribution are included. However, the error-rate on Test Set 3 remains high, and using the duration actually increases it a little. This may be because our duration model was from a different match, with a different set of commentators, a different umpire, and under different conditions in which, for instance, the duration of the crowd noise may have been rather different.

	Training	Test1	Test2	Test3
GMM+Vit.+F-LM M-LM+E-LM	8.66%	17.11%	23.10%	31.38%
+duration	7.71%	15.76%	22.20%	31.67%
+pitch	7.05%	14.89%	19.68%	26.95%

Table 4. Frame error rate using the information of event duration and pitch distribution

6. SUMMARY AND DISCUSSION

In this paper, we have presented a technique for classifying audio events using a hierarchical structure that integrates low- and high-level models of the events. We have also integrated duration and pitch information into the classification process. Our initial results are encouraging, giving relative improvements in the frame error-rate of the order of 50% when compared with labelling using GMMs alone. The results show that using a low-level "language model" of frame events is the most powerful technique, and the extra gain from using the a "language model" of frame events is small. However, we have not yet experimented with varying the "grammar factor" of this language model. We have also shown using duration and pitch information can provide significant improvements in accuracy.

Our future work is to look at the issue of how to balance the probabilities from the different language models used here, and how to integrate in a more effective way the contributions of the duration and pitch information. We are also considering replacing the GMMs with ergodic HMMs in order to provide more accurate initial frame labelling.

ACKNOWLEDGMENT: This work was supported under a UK Engineering and Physical Sciences Research Council Grant number EP/F069626/1.

7. REFERENCES

[1] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, Iain McCowan, and Guillaume Lathoud, "Modeling individual and group actions in meetings: A two-layer hmm framework," in *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, Nagoya, Japan, 2004, pp. 117–124.

[2] Alfred Dielmann and Steve Renals, "Automatic meeting segmentation using dynamic bayesian networks," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 25–35, 2007.

[3] Yihong Gong, Mei Han, Wei Hua, and Wei Xu, "Maximum entropy model-based base baseball highlight detection and classification," *Computer Vision and Image Understanding*, vol. 96, pp. 181–199, 2004.

[4] Jinjun Wang, ChangSheng Xu, and Engsion Chong, "Automatic sports video genre classification using pseudo-2d-hmm," in *Proceedings of the 18th International Conference on Pattern Recognition*, 2006, pp. 778–781.

[5] Nam Nguyen and Yunsong Guo, "Compariosns of sequence labeling algorithms and extensions," in *Proceedings of International Conference on Machine Learning*, June 2007.

[6] Xian Qian, Xiaoqian Jiang, Qi Zhang, Xuanjing Huang, and Lide Wu, "Sparse higher order conditional random fields for improved sequence labeling," in *Proceedings of International Conference on Machine Learning*, 2009.

[7] T. Hain and P.C. Woodland, "Modelling sub-phone insertion and deletion in continuous speech recognition," in *Proceedings of International Conference on Speech and Language Processing*, 2000, pp. 172–176.