SFERANET: AUTOMATIC GENERATION OF FOOTBALL HIGHLIGHTS

Vincenzo Scotti, Licia Sbattella and Roberto Tedesco

DEIB, Politecnico di Milano, Milano, Italy

{vincenzo.scotti, licia.sbattella, roberto.tedesco}@polimi.it

ABSTRACT

We present a methodology for automatic generation of football match "highlights", relying on the commentator voices and leveraging two multimodal NNs.

The fist model (M1) classifies sequences and provides a representation of such sequences to be elaborated by the second model. M2 exploits M1 to decode unbound streams of information, generating the final set of scenes to put into the match summary.

Raw audio, along with transcriptions generated by an ASR, extracted from 369 football matches provided the source for feature extraction. We employed such features to train M1 and M2; for M1, the feature streams were split in sequences at (nearly) sentence granularity, while for M2 the entire streams were employed. The final results were promising, especially if adopted in a semi-automatic, real-world video pipeline.

KEYWORDS

Neural Networks, NLP, Voice, Text, Summarisation

1. INTRODUCTION

There are many motivations behind this project. First of all, living in a modern era were people have such easy access to information everywhere at any time has made them willing to be constantly and immediately updated. In this sense, sport fans have become more and more hungry; this can be easily seen by the amount of web sites updated with the results of each match in real time, the streaming services to watch the events, and the video sharing platforms.

However, it would require a huge amount of time to watch all the events, even for a single sport. Sport highlights, which are becoming more and more popular and heavily used by broadcasting companies, provide a recap of the most exciting parts of a sport event. It is a convenient way for knowing what happened in, for example, a round of your preferred football championship.

So far, such highlights are created by manually edit the raw video recordings, but we think there is room for improving the current video pipeline, by means of a tool that speeds up the process. In particular, we envision a semi-automatic pipeline where a tool generates a first version of the highlights, while the human editor only needs to refine them.

The aim of SFERAnet (Selection of Football Events by Recorded Audio) is to train a Neural Network (NN) able to identify top moments inside a football match through the analysis of the commentators' voices. In practice, the idea is to detect the segments where the speakers show *excitement*.

Natarajan Meghanathan et al. (Eds) : CSEIT, CMLA, NeTCOM, CIoT, SPM, NCS, WiMoNe, Graph-hoc - 2019
pp. 101-116, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91309

We designed two models: one able to perform sequence classification and one, encapsulating the former, able to deal with the entire event stream and giving a continuous output on the importance of the sequences of the event. Starting from that importance measure it would be possible to extract what should belong to the final event highlights.

We didn't make use of video-based features, like scoreboard graphics or sophisticated scene recognition, as the former depends on the broadcasting network and the latter requires a huge corpus.

SFERAnet is thought to be used inside a semi-automatic video pipeline, where a human editor refines the video generated by the tool.

2. RELATED WORK

Our work is based on "excitement recognition" through speech analysis, which is conceptually similar to the common task of emotion recognition. Moreover, we also leveraged literature on automatic detection of sport highlights. In the following we present some relevant papers on both topics.

2.1. Automatic Emotion Classification from Speech

Focusing our analysis to NNs, we found that the approach evolved considerably through time, especially in the last few years. End-to-end NN solutions were first brought by a work [1] proposing a simple densely-connected NN with three hidden layers to transform acoustic features –computed from utterances sub-splits– into sequence of probability distributions over the target emotion; then, probabilities were aggregated into utterance-level features using simple statistics (such as maximum, minimum, average etc.) that an Extreme Learning Machine (ELM) model used to classify the utterances.

A following work [2] proposed an improvement replacing densely-connected layers with recurrent ones; in particular, they used Long Short-Term Memory (LSTM) layers. However, they continued using local-probability aggregation into a global features vector, and Extreme Learning Machines (ELM) on top of them, to perform the classification task, as in [1].

The use of simple and naïve aggregation functions and ELMs resulted not only in a drawback for these two approaches, but also in criticism; another work [3] aimed at getting rid of the drawbacks discussed above by applying fully end-to-end pipeline without handcrafted parts in the middle. The proposed solution consisted, again, in an LSTM architecture with Connectionist Temporal Classification (CTC) approach [4] to assess the class, which proved to be useful also to deal with the different lengths of the utterances.

The last work we cite [5] used both acoustic and *linguistic* features (i.e., features coming from the textual transcription of the speech). The author compared three different models: *audio-only*, *text-only*, and *mixed*. This work, which reported an overall accuracy of 74.3% for the mixed model on the IEMOCAP corpus [6], clearly showed that a multimodal approach provides the best accuracy. For this reason, we decided to follow the same approach.

2.2. Identification of Sport Event Highlights from Speech

A first attempt proposed a system for automatic detection of baseball highlights [7], based solely on audio analysis of the commentator. The hypothesis that guided this work was that high correlation exists between speaker's voice excitement and relevant events. However, since not all events could count on the presence of speech into the background, they also considered a

baseball-specific feature: the presence of a baseball hit in the audio track. So, authors considered two distinct SVM models: identification of excited speech and identification of baseball hit candidates. Then the results from these two models were fused to provide a final estimation of the probability that the analysed segment was exciting. Eventually they reported an overall accuracy of 75%.

Another work [9] proposed an audio-based model for tennis, combining long- and short-term features. Authors presented a cascaded architecture composed of two levels. The first one, worked only on short-term features using a SVM with Radial Basis Function (RBF) kernel; on top of them a Bayesian inference model combined the results from both and generated the prediction for the considered window. The second level took both long-term features and class predictions from the first one. For both audio classifiers at the base of the model, only Mel Frequency Cepstral Coefficients (MFCC) vectors were considered as input, while the output classes were: silence, applause, and speech. Authors reported precision of 98% and recall of 96%.

In [10], instead, authors proposed a system architecture based on Piecewise Gaussian Modelling (PGM) and NNs to detects highlights, but still working only on the audio signal. In this work they tried to detach from the energy-based features, like in [7], by employing the Mel Frequency Spectral Coefficient (MFSC) representation of the audio signal as a short-term feature. The resulting feature vectors are combined through PGM to achieve a long-term description of non-overlapping, fixed-size frames of the Mel Spectrogram that are classified by the NN as “action” and “no-action” (i.e. the labels they considered for the scenes into the highlights). Authors underlined two key points about their system: it only needs a few seconds of audio samples to build the classifier, and the architecture, being based only on audio features, can be effectively employed in different sports by providing results for tennis and football. In fact, they achieved a precision of 87.2% and a recall of 97.6% in detection of highlights for tennis, and an average precision of 86.7 % in the three football matches used for tests.

2.3. Identification of Sport Event Highlights from Speech and Video

One of the first systems leveraging both audio and video clues was presented in [8], where authors proposed an audio-visual framework for sport event detection. In their work, authors pointed out some useful information, in fact they noticed how sport-specific approaches typically yielded successful results within the targeted domain because of the dramatic variances in commentary styles for different sports. However, their intention was to build a general model able to work with different sports, and this is why their data set was composed of events from football, rugby, and Gaelic football. The solution they proposed was based on a SVM classifier able to separate eventful and non-eventful sequences; for this goal the SVM took as input an aggregated features vector composed by: crowd image detection, speech band audio activity, on screen graphics tracking, motion activity measures, and field line orientation. Authors reported in the case of Gaelic football an event retrieval ratio of 97%, this was the best achieved scored among their classifiers

Other relevant results in this field came from a work [11], where authors focused on visual features. The proposed system was composed of two main blocks: an unsupervised framework for event decomposition based on Hidden Markov Model (HMM), which performs diarisation of the clips (i.e. segmentation and clustering) iteratively, and a subsystem for detection of highlights, which takes out the classification task on the events to discriminate between *highlight* and *non-highlight*, based on a Linear SVM. The system worked with “easy-to-extract, low-level” visual features: the Colour Histogram (CH) and the Histogram of Oriented Gradients (HOG), which were projected to a lower dimensional space through Principal Component Analysis (PCA) in order to avoid the curse of dimensionality. The authors trained and tested the system using video clips from cricket matches (they were provided with 14000 clips that they split in half for this

purpose) and explored the results when features were considered singularly and together, achieving an equal error rate of 12,1% when using both.

More recent results [12] proposed a system for detection of rugby highlights, based on detection of acoustic events. In particular they built a multi-stage classifier, that considered two acoustic events to perform the classification task: commentator's excited speech and referee's whistle. In the proposed model a first-stage classification is applied to detect from the input audio features, then excited speech detection or whistle detection are performed; at this point, time stamps of positive classification from the second stage are stored in a buffer that is later scanned to detect if a minimum number of relevant frames are present in a fixed temporal window. Then the window is extended to cover all the relevant events for that particular scene. All the three classifiers were built using GMM, and the selected audio features were MFCC, together with their first order derivatives; in this case the reported precision was 93.4% and the recall 97.1%.

The following year, with the spread of eSports championships, a video-based highlight detector for Multiplayer Online Battle Arena (MOBA) games was proposed [13]. The author proposed various solutions for frame-wise classifiers based on CNN and RNN, considering both single and cascaded architectures, and different shapes for the output; in fact, the data set was tagged considering four different levels of highlight, starting from *non-highlight* up to *maximum relevance*. The peak performances were achieved, mostly, considering only a binary output: one of the considered games reported a precision of 83.2% and a recall of 86.3%. A point to stress out about this model is that it was designed to work with real-time video streams.

Finally, we mention H5 [14], a multimodal system for extraction of highlights from sport videos, based on sport-independent excitement measures (although in the paper only Tennis is analysed as a case study). The H5 system employed excitement markers, coming from different modalities, to score the scenes of the match; in particular, authors distinguished between audio- and text-based markers, visual markers, and game analytics. Audio-based markers were extracted through a SVM built atop deep features (coming from a Deep Convolutional Neural Network used for audio classification purposes) to classify *crowd cheer* and *commentator tone excitement*; moreover, the commentator tone was complemented by a text-based marker that matched the transcription against a dictionary of expression indicative of excitement. Visual markers, instead, were computed through two classifiers, one for *player reaction* (scenes were a player was celebrating) and the other for *facial expression* (categorized in *aggressive*, *tense*, *smiling*, and *neutral*), both obtained fine-tuning pre-trained Deep Convolutional Neural Networks for image classification. Game analytics, instead, referred to Tennis specific information; in fact, since not every point in the match has equal relevance, a side court statistician provided information distinguishing between different points (e.g. *volley winner, smash winner, match point*, etc.). The sub-models composing H5 were trained separately on manually tagged audio and video clips to extract the markers, then a separate fusion model was trained to classify the proposed clips from the markers and discriminate the highlights. To test H5 a group of users was asked to rank from 0 to 5 their interest in randomly selected clips from those proposed by H5, scores was averaged to compute the precision of the system that resulted to be 92.68%.

2.4. Comments

The results obtained by the presented works are really good; nevertheless, in many cases such systems took advantage of particular visual features, for example enabling scoreboard graphics tracking as in [8], which represented a strong aid (but made the system dependent on the broadcast network-specific graphics). In other cases, like [13], [14], that leverage DCNN to perform the analysis of the visual input, a higher computational capacity is required, not to mention the necessity of a large amount of data. In [14] authors tried to cope with this problem by fine tuning pre-trained models, but the demand of computation power remained high since the

transferred models were still huge and they still needed to build a personalized data set "by hand" to extract their markers.

Keeping on with [14], there are two other key problems to point out: they were given access to game analytics provided by side court statistician that provided information in real time about the scored points (such information is hardly available, especially considering different sports) and they had the financial capabilities to pay a group of users to score their highlights.

Moreover, as in the case of [7], metrics were computed "by hand", in the sense that a human operator compared the resulting highlights on a *small test set* with the expected output, to circumvent errors due to misaligned highlights. Our corpus was composed of thousands of samples, making it impossible to follow the same approach. Therefore, we followed the usual cross-validation approach, without human intervention.

Another particular case is that of [9], [10], where the crowd remains silent for the whole game, except to applaud immediately after a point is scored; so, highlights were basically located by the occurrence of applause. Using such a sport-specific clue wasn't possible in our case. Actually, in [10] an alternative for football was proposed: take advantage of crowd's noise together with commentator's excitement. This choice resulted, in the authors' own words, in an approach "extremely sensitive to the spectators' and commentators' behaviours" for both of the analysed sports. Moreover, in our dataset the recordings of the crowd weren't provided as a separated audio channel, and it was only possible to hear them in the background of the commentators' voices, making it very difficult to leverage such information.

In [11], instead, two other problems were introduced. The former was that highlights were given a fixed definition (according to cricket terminology, the highlights were defined as video clips corresponding to either a *4-run*, a *6-run*, or a *wicket*) so what they actually produced was a system capable of identifying these exact events and nothing else; on the contrary, we wanted to avoid to impose a fixed rule to define the highlights. The latter was that even if the proposed system carried out event discovery within a clip, all the information from the events within that same clip were employed for classification, so the system still relied on previously cut clips of fixed length; instead, we wanted to provide a system capable of finding also the cut points of the scenes to put into the highlights.

Finally, authors of [12] employed a small data set, which was tagged manually to identify as "important" everything that they expected to trigger their system. This led, indeed, to good results, but they were a consequence of this ad-hoc choice. Differently, our corpus was based on highlights generated by professional video editors.

Summing up, the system we are presenting leverages only speech and textual features from the match commentary, which can be considered as sport-genre independent; in this way, our system can be easily ported to other sports. Moreover, as shown in previous sections, various attempts in the past years proved the presence of a relationship between the excitement in the speaker voice and the importance of the related scene; this further convinced us to follow the same approach and leverage audio features. Finally, the choice of such features results in a smaller model, easier to train and faster to run.

3. DATA SET

This section presents all the information about the data employed to train the NNs.

3.1. Provided Data

Data come from 369 football matches of the 2017-18 Italian "Serie A" championship; each video recording come with the corresponding hand-crafted highlights. Each match highlights were composed of about 20 short sequences (we call them *scenes*); see Figure 1. As the commentators' audio tracks contained the chattering and interviews before and after the match, each video was manually searched for finding the actual starts and end of the two halves of the match.

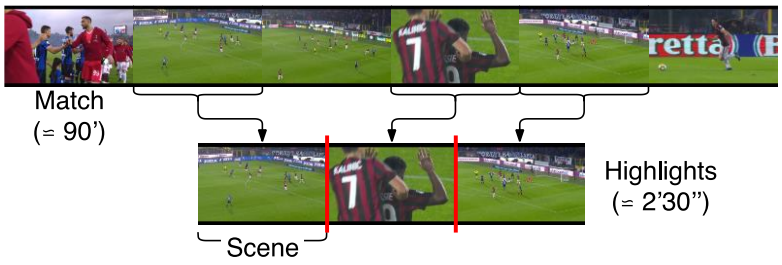

Figure 1. Match video recording and its "highlights" scenes.

3.2. Label Generation

No proper tagging of the original data was provided. To deal with this problem, we realized a tool based on perceptual hashing of images. In practice, given the video of a match and the corresponding highlights, they were both down-sampled to a grey-scale, 160 × 90, 10 fps streams. Subsequently the pHash algorithm [15] was applied to each frame.

Then, the hashes of consecutive frames belonging to the same scene in the highlights were grouped together in temporal order. In this way not only the entire video scene from the highlights could be searched at once, but the results come out to be more robust since the similarity score was averaged on the entire scene. To split the highlights into scenes the similarity score between each frame and its successive was computed; in this way a drop under a fixed threshold could identify a scene change.

Each group of hashes, representing a scene, was searched computing the average similarity score against a sliding window, of the same length of the currently searched hash group, scanning the whole match. The starting frame of the window corresponding to the highest similarity score was retrieved. Once the time markers for each scene had been identified, close segments were joined; this step was necessary because sometimes either the highest similarity score didn't lead to a perfect alignment or a part of the scene had been cut away during the editing of the summary video.

At the end, we divided and tagged the match segments:

Relevant: segments showing scenes used to compose the highlights.

Non-relevant: here are all discarded segments of the match.

3.3. Corpus Internal Structure

The corpus is composed of audio recordings of the match commentaries and, through an Automatic Speech Recognition (ASR), the corresponding transcriptions. Using the Google Speech-to-Text API permitted to obtain word-level timing alignment and speaker differentiation.

Each match contained about 2h of data but only the in-game spoken parts were considered. In this way the total amount of recordings resulted to be 640h, divided in 13h of Relevant segments and 627h of Non-relevant segments. Because of this unbalance, samples from the Non-relevant class were randomly selected in order to obtain a balanced data set so that the NN won't be badly influenced; Moreover, the "subsampling" was performed file-wise so that from each match the same amount of segments per class could be used.

In this case the term *sample* refers to the constitutive element of the data set: a scene containing audio and textual data, aligned, and coupled together. To cut the Non-relevant segments we employed a heuristic algorithm that grouped consecutive spoken parts, identified through Voice Activity Detection (VAD), in clusters of, approximatively, the same length of the Relevant ones.

These sentences were grouped into the development set, composed of segments coming from the first 50 matches, which helped to identify possible network models and hyper-parameters, and the actual data set, which used all the 369 matches segments to train, validate and test the most promising configurations and find the best one.

The content of the data sets, in terms of number of samples and duration, is reported in Table 1.

Table 1. Corpus information.

	Number of samples			**Duration (sec)**	
	Total	**Non-relevant**	**Relevant**	**avg**	**std dev**
All available	390932	384549	6383	3.66	2.15
Data set	12766	6383	6383	7.40	4.50
Dev set	1832	916	916	7.43	4.68

The corpus is composed by the voice of 20 different male speakers. Having a wide, different speaker presence in the data set is critically significant since it helps the network to avoid being dependent on the specific speaker's behaviour, especially for speaker-dependent features.

4. DATA PREPROCESSING AND FEATURE EXTRACTION

The raw audio signals sampled at 48 kHz were the starting point from which the input features were extracted to feed the NN-based models, this extraction process required some critical preprocessing steps that consisted, mostly, in noise suppression and downsampling; moreover in many cases, depending on the feature typology, some additional post-processing, like outlier deletion and filtering, was also required.

4.1.Audio Preprocessing

In the commentators' audio files, it was possible to hear the crowd cheering in the background. Since this noise was frequently overlapping with the voice signal to analyse, the RNNoise tool [16] was employed to get rid of it.

To reduce the amount on information to be processed, the audio tracks were down-sampled at 16kHz and the features were computed with a 20 ms wide sliding window, with a hop size of 10 ms, obtaining 100 samples per second.

Then, the preprocessing workflow executed VAD and speaker diarisation, whose results were later used for the computation of the features. In particular, results from the latter were employed to reach speaker independence. Anyhow, their use will be better explained in the following section.

4.2. Selected Features

Even though it's a common practice to leave DNNs learn the features by themselves, this approach may lead to sub-optimal solutions and incredibly complex models with subsequent waste of computational resources, as suggested by the authors of RNNoise. For these reasons, the classifiers were trained on a set of carefully selected, pre-computed features that already proved their relevance. In particular, the features we used can be classified into three groups [3]:

Prosodic. These features describe voice intonation, rhythm, and stress; we used: pitch, intensity, harmonicity, jitter, shimmer (along with their first- and second-order derivatives), chroma, silences (pauses), short-term energy with its entropy, and syllabic rhythm.

Acoustic. These features describe the spectral properties of voice; we used: MFCC, Mel bands decomposition, centroid, spread, entropy, flux, roll-off, and zero-crossing rate.

Linguistic. These features describe the semantic information contained in speech; we used word embeddings.

Prosodic and acoustic features were selected because of their correlation with perceptual aspects of the signal [17], [18], for example the pitch expresses the sentence intonation.

Apart from these features, another one represented the relative time position of the analysed segment with respect to the entire recording (the very beginning and the end of the match are very likely to be put into the highlights).

All the computed features were post-processed before being fed to the NNs, in particular we adopted a speaker-wise approach in order to obtain speaker-independent features. The post-processing steps were: standardisation, making the values of each feature in the data have zero-mean and unit-variance, outlier trimming, silent segments zeroing, and signal smoothing.

5. MODELS

Our model is actually composed of two sub-models:

- M1: for scene classification.
- M2, incorporating M1: for stream decoding, producing a continuous classification output.

The reason behind this choice stands in the structure of the former model as well as in the experiences coming from other projects. In fact, the main processing element of M1 stands in the BLSTM layer, that provides a powerful tool to analyse a scene by scanning it from start to end and vice-versa at the same time. However, as a drawback, having to deal with too long scenes, as in the case of an entire football match audio features stream, will most certainly produce poor results since the portions analysed by the forward and the backward LSTMs will be too uncorrelated.

The proposed solution for this particular problem is to train M1 separately such that it's able to *classify a single scene of known length*; after that, M2 can be trained applying transfer learning from M1, that will be used to provide a useful windowed representation. To be more detailed, the second model will perform continuous stream labelling from feature windows computed from the transferred part of the first classifier and will use a mono-directional recurrent layer to add the context from the previously analysed windows.

5.1. M1: Multimodal Scene Classifier

M1 is designed to classify a scene of variable but known length (see Figure 2); it was trained on short video scenes, namely less than a minute, however it can ideally work with unbounded ones even though performances are not ensured to be the same.

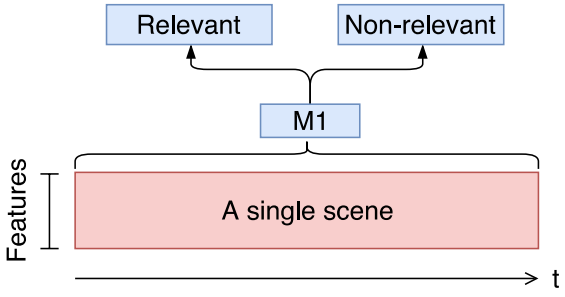

Figure 2. M1: scene classification.

As shown by Figure 3, this classifier takes the raw features of a scene and feeds them to a one-dimensional, time-distributed convolutional layer with dilation, immediately followed by a one-dimensional, time-distributed, max-pooling layer. Then, the intermediate results from the input layers are passed to a BLSTM provided with an internal attention mechanism; the BLSTM layer produces a continuous output that is weighted by the output of the attention mechanism. These weighted values are then summed up along the time to have a compressed representation of the entire scene, and the sum is scaled using a logarithmic function. The resulting intermediate representation of the entire scene is then passed to two subsequent fully-connected layers before arriving to the softmax layer with two output that represents the probabilities to belong to one of the two classes.

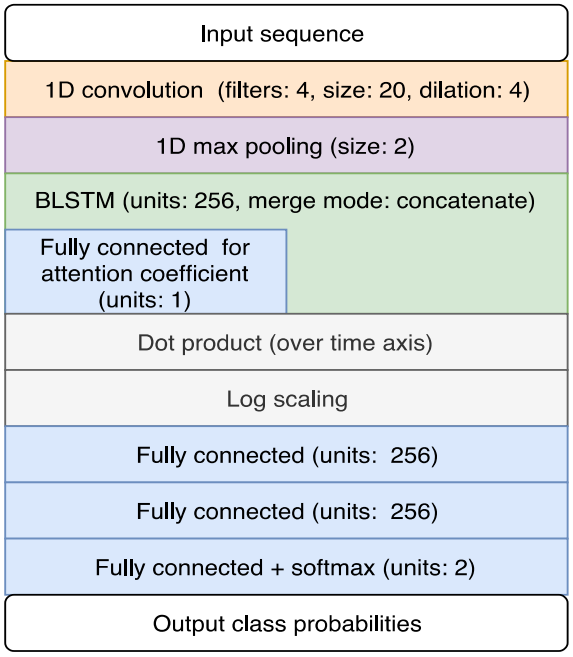

Figure 3. M1: scene classifier DNN.

5.2. M2: Multimodal Stream Decoder

M2 is designed to classify a scene of variable is designed to decode an entire stream providing a continuous classification output; it was trained on streams corresponding to entire matches.

As shown by Figure 4, this classifier takes the raw features stream as input, then slices it using a fixed-size sliding window of 7.5s with a 3.75s hop. Windows are fed in sequence to an internal time-distributed model, realised using M1, which generates an internal representation of the entire window content. These intermediate representations of the widows are then passed in sequence to an LSTM that will provide some sort of "context" among successive windows.

The continuous output of the LSTM is further elaborated by a time-distributed, fully-connected layer before the time-distributed softmax layer accomplishes the decoding task. This last layer associates the probability to belong to one of the two classes to each of the windows generated at the beginning of the pipeline. We then applied a threshold of 0.5 to identify the start and end points in time of the Relevant segments.

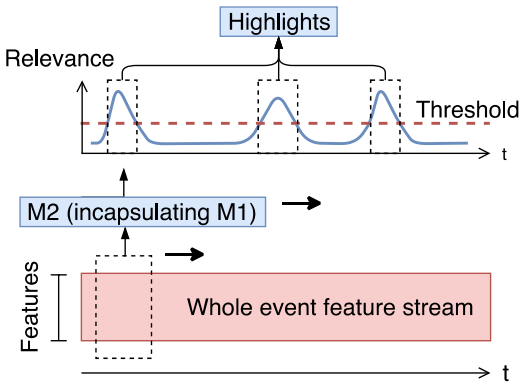

Figure 4. M2: stream decoding with sliding window.

Figure 5 shows the structure of the M2 classifier. Notice that the size of the sliding window is a parameter to be decided at "run time", is not part of the definition of M2, and does not constraint in any way the length of the retrieved scenes.

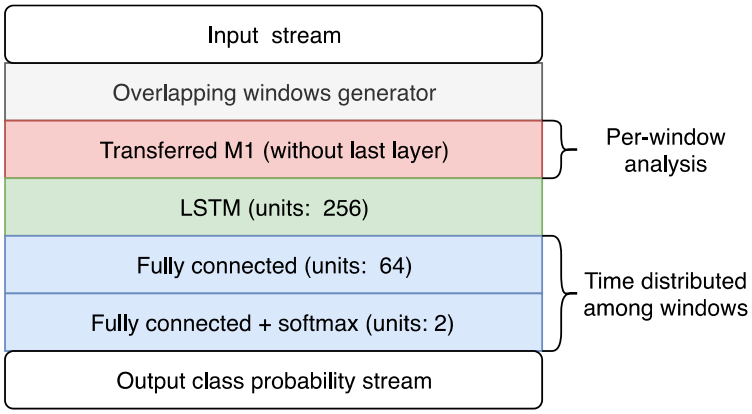

Figure 5. M2: stream decoder DNN.

As a final remark, our approach shows interesting features:

- It won't be necessary to train the entire network of M2 from scratch; in fact, thanks to transfer learning, only the top portion of the network requires training.

- Size and hop of the sliding window fed to M2 can be modified, within certain limits, without having to train the M1 network from scratch, this is due to that fact that the BLSTM layer in M1 is designed to deal with and trained on variable length scenes.

5.3. SFERAnet

Figure 6 shows SFERAnet, inside a hypothetical semi-automatic video pipeline for generation of highlights. The pre-processed speech audio is passed to an ASR and enters, along with the transcription the SFERAnet models. The result is a set of cut points (i.e., time instants where the video stream should be cut to extract the relevant scenes). Then, some video editing tool (for example, FFmpeg) could be used to generate the proposed highlights. Finally, a human expert composes the final version by means of her/his usual video editing tools.

The proposed solution for this particular problem is to train M1 separately such that it's able to classify a single scene of known length; after that, M2 can be trained applying transfer learning from M1, that will be used to provide a useful windowed representation. To be more detailed, the second model will perform *continuous stream labelling* from feature windows computed from the transferred part of the first classifier and will use a mono-directional recurrent layer to add the context from the previously analysed windows.

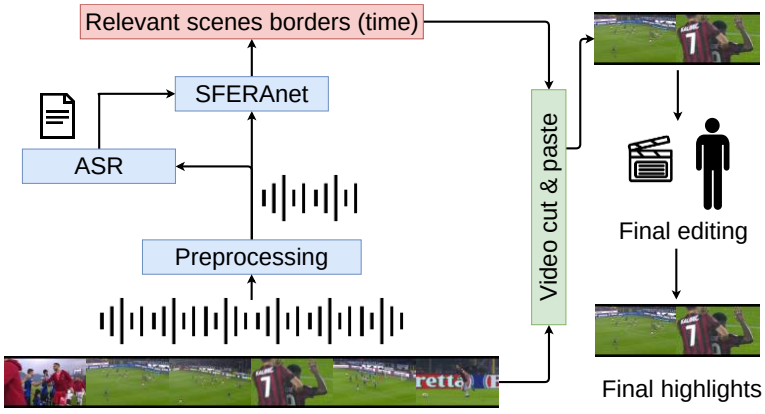

Figure 6. The SFERAnet semi-automatic pipeline.

6. TRAINING AND VALIDATION

This section will deal with the description of the training and validation process to find the best architecture.

6.1. Approach

The procedure to find the best model followed the same steps for both models. M1 was trained and validated on the samples of the data set we described in Table 1; for M2, instead, we considered as a sample the entire feature stream coming from a whole match.

The first step consisted in a grid search vowed to find the best model structure; at this stage the objective was to obtain the main structure of the model, without refining it, using a development set obtained by random-subsampling the data set.

The second step consisted in the refinement of the hyper-parameters of the best model found through the grid search, again on the development set. Differently from the previous stage, in this case there was a tree search (to lower time complexity, although at the cost of finding a sub-optimal model).

In the last step, the most promising models were compared using the results from the training on the entire data set.

In each of the presented steps, the evaluation of the model was obtained through a 10-fold cross-validation; in this way a more robust estimate of the performances could be obtained. In the train phase relative to each fold, a further split of the train set was created to be used as a validation set.

Training was performed using categorical cross-entropy as loss function, RMSProp as optimiser, and adopting the early stopping strategy. As performance metrics we computed Accuracy (using it also as a reference for early stopping), Precision, Recall, F1, Specificity, and AUC.

To deal with the class unbalance inside the data set, we considered two different approaches, depending on the model. For what concerns M1, we randomly sub-sampled the class of Non-relevant to get an equal number of scenes. For M2, instead, loss and Accuracy were weighted differently depending on the class, so that an error on the Relevant segments would be 60 times that of the Non-relevant class; the choice of that weight was done in order to reflect the available hours of recordings of each class inside the corpus.

6.2. Results

Table 2 shows quantitative values for both models. For what concerns M1, considering the best model, and in particular the results from the single best fold, it showed a high Precision. This means that M1 is particularly good in discarding the Non-relevant scenes, making it suitable for a real-world video pipeline. Moreover, it is important to stress how, with a balanced data set, Accuracy, F1 and AUC –which are used as global measure considering all the classes– show good values.

Table 2. Best results achieved by M1 and M2. The reported results are these of the models with the highest *cross-validation* (weighted) *accuracy* score.

Metric	**Model**					
	M1			**M2**		
	avg	**std dev**	**best**	**avg**	**std dev**	**best**
Accuracy	0.811	0.025	0.846	0.488	0.046	0.588
Weighted accuracy	-	-	-	0.682	0.020	0.715
Precision	0.884	0.035	0.900	0.032	0.003	0.038
Recall	0.719	0.063	0.778	0.852	0.033	0.822
Specificity	0.903	0.037	0.914	0.481	0.047	0.583
F1 Score	0.791	0.036	0.835	0.062	0.005	0.073
AUC	0.894	0.014	0.918	0.775	0.021	0.797

M2, instead, showed way lower scores with respect to M1. However, these scores are to be taken with a grain of salt. In fact, we found two different error categories: *model specific* and *summarisation specific*.

The model-specific errors are due to the fact that the output probability stream may be noisy around the classification threshold; in this case the problem may be fixed improving a post-processing phase. Moreover, the output probability stream may rise above the classification threshold before it is done in the target scene, and/or similarly may fall down after it, as depicted in Figure 7 (left); as scores are computed for each instance of the sliding window (i.e., every few milliseconds), even if the retrieved scene *contains* the correct one, several window instances fall outside the right interval and the computed scores are badly affected. Another typical scenario is depicted in Figure 7 (right), where a single retrieved scene contains multiple correct ones. Once again, the scores could be very low even if the model prediction is substantially correct.

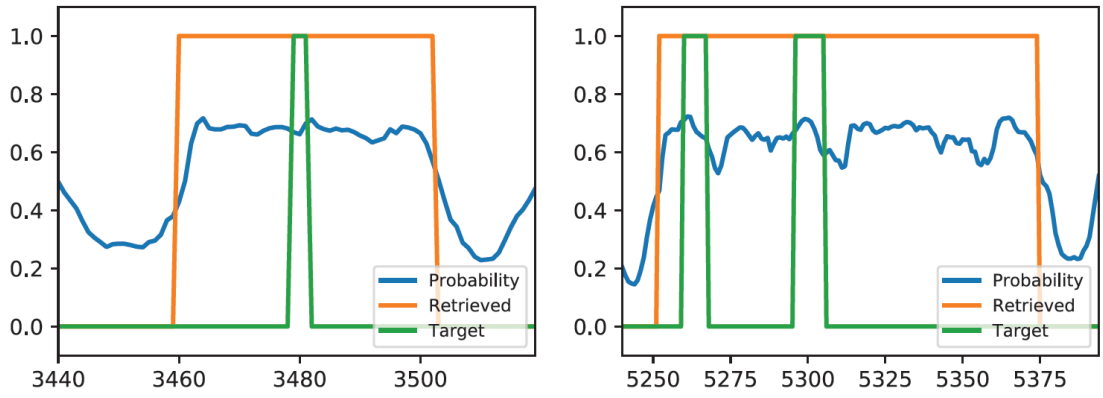

Figure 7. Output of M2 (blue), extracted scene (orange), and ground truth scene(s) (green).

The summarisation-specific errors are due to the fact that there is no "correct" metric to assess the goodness of a summary. In fact, scene cut points are somewhat arbitrary and the selection of the scenes is, to some extent, arbitrary: usually there are more relevant scenes than the ones found in the final highlight; such scenes in "excess" are cut due to time limitations (highlights shouldn't last more than 3 minutes) but are not Non-relevant per se. For that reason, the figures we report in Table 2 are based on the usual metrics computed comparing samples in a classification task (where a sample is a decoded window of the match feature stream).

As a further validation step for M2, one should appeal to human evaluation, as some research papers we cited did. However, on one hand our corpus was too big to allow for this solution; on the other hand, a human evaluation is subjective and, in our opinion, should be avoided.

Unfortunately, in this way the problem of finding remains open but, on the other hand, it is a well-known issue even in the much more mature field of text summarisation [19]. As a final remark, better metrics could be very useful for improving the train of the model.

7. CONCLUSIONS AND FUTURE WORK

The results that are not easy to be evaluated. If M1 proved good in selecting Relevant scenes, M2 is probably not manure enough. However, in a real-world video pipeline, SFERAnet will be just a tool for a human operator. For her/him, cutting a useless scene (false positive) would be easier than add a missing scene (false negative). From this point of view, the Recall of M2 is not bad and thus SFERAnet could be actually useful, as long as it is employed in a semi-automatic pipeline.

As a future improvement, assuming to get a bigger corpus, we aim at testing more complex architectures, like GANs, which proved very powerful tools for "generation via emulation" and thus could produce more human-like highlights.

Finally, we expect to carry out some experiments on the field, by generating the highlights of matches "unseen" by SFERAnet and observing users' reception.

REFERENCES

[1] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, Sep 2014, pp 223–227.

[2] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden (Germany), Sep 2015, pp 1537–1540.

[3] V. Chernykh and P. Prikhodko, "Emotion recognition from speech with recurrent neural networks," in *CoRR*, arXiv preprint arXiv:1701.08071, Jan 2017.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in Proceedings of the 23rd International Conference on Machine Learning, New York (NY, USA), Jun 2006, pp. 369–376.

[5] J. M. Origgi, "PATHOSnet: parallel, audio-textual, hybrid organization for sentiment network," Master's thesis, Politecnico di Milano, 2018. [Online]. Available: http://hdl.handle.net/10589/143008

[6] C. Busso, M. Bulut, C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: interactive emotional dyadic motion capture database," in *Language Resources and Evaluation*, vol. 42, no. 4, Nov 2008, p. 335.

[7] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in Proceedings of the Eighth ACM International Conference on Multimedia, New York (NY, USA), Oct 2000, pp. 105–115.

[8] D. A. Sadlier and N. E. O'Connor, "Event detection in field sports video using audio-visual features and a support vector machine," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, Oct 2005, pp. 1225–1233.

[9] B. Zhang, W. Dou, and L. Chen, "Combining short and long term audio features for tv sports highlight detection," in *Advances in Information Retrieval*, M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 472–475.

[10] H. Harb, L. Chen, "Highlights detection in sports videos based on audio analysis," Oct 2009.

[11] H. Tang, V. Kwatra, M. E. Sargin and U. Gargi, "Detecting highlights in sports videos: Cricket as a test case," in 2011 IEEE International Conference on Multimedia and Expo, Barcelona, Jul 2011, pp. 1-6.

[12] A. Baijal, J. Cho, W. Lee, and B.S. Ko, "Sports highlights generation based on acoustic events detection: A rugby case study," in 2015 IEEE International Conference on Consumer Electronics, Jan 2015, pp 20–23.

[13] Y. Song, "Real-Time Video Highlights for Yahoo Esports," Nov 2016.

[14] M. Merler, D. Joshi, K.C. Mac, Q. Nguyen, S. Hammer, J. Kent, J. Xiong, M.N. Do, J.S. Smith and R.S. Feris, "The Excitement of Sports: Automatic Highlights Using Audio/Visual Cues," in 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Jun 2018, pp. 2520-2523.

[15] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *Ismir*, vol. 32, Jan 2002, pp. 107–115.

[16] J. M. Valin, "A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement," in *CoRR*, arXiv preprint arXiv:1709.08243, Sep 2017.

[17] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2009.

[18] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," in *CUIDADO IST Project Report*, vol. 54, Jan 2004, pp 1–25.

[19] N. Schluter, "The limits of automatic summarisation according to ROUGE," in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia (Spain), Apr 2017, pp. 41–45.

AUTHORS

Vincenzo Scotti received the B.Sc. and the M.Sc. in Computer Science and Engineering from the Politecnico di Milano respectively in 2016 and 2019. He is now a PhD student in Computer Science and Engineering at the Politecnico di Milano.

Licia Sbattella Ph.D. in Computer Science, Bioengineer and Clinical Psychologist. She is Associate Professor of "Natural Language Processing" and "Personality, Team building and Leadership" at Politecnico di Milano. Since 2003 she is the Delegate of the Rector for persons with disability and psychological difficulties. She is member of the Steering Committee of UNG3ict and cooperates with the International Association of Universities (IAU) and with the Pontificia Accademia per la Vita.

Roberto Tedesco earned a M.Sc. in Computer Science, in 2001, and a Ph.D. in Computer Science, in 2006, both at Politecnico di Milano. He is contract researcher at Multi Chance Poli Team, Politecnico di Milano. His research interests are: Natural Language Processing, assistive technologies, user profiling and service customization, and e-learning.