Abstract—We present an approach to automatically generating verbal commentaries for tennis games. We introduce a novel application that requires a combination of techniques from computer vision, natural language processing and machine learning. A video sequence is first analysed using state-of-the-art computer vision methods to track the ball, fit the detected edges to the court model, track the players, and recognise their strokes. Based on the recognised visual attributes we formulate the tennis commentary generation problem in the framework of long short-term memory recurrent neural networks as well as structured SVM. In particular, we investigate pre-embedding of descriptive terms and loss function for LSTM. We introduce a new dataset of 633 annotated pairs of tennis videos and corresponding commentary. We perform an automatic as well as human based evaluation, and demonstrate that the proposed pre-embedding and loss function lead to substantially improved accuracy of the generated commentary.

I. INTRODUCTION

Recent advances in computer vision, natural language processing and machine learning have led to a dramatic increase of interest in integrated modelling of vision and language. In particular, the methodology for generating natural language descriptions for images [1], [2], [3] and videos [4], [5], [6] has made significant progress in the last few years. The state-of-the-art methods employ convolutional neural networks (CNNs) [7] to encode visual information, and long short-term memory (LSTM) recurrent neural networks (RNNs) [8] to decode the information into sentences. In these methods, tokens (words in the sentences) are represented as one-hot vectors and back-propagation is driven by perplexity or negative log likelihood loss that is based on estimated probability distribution of the tokens. As a result, the potential correlation between the tokens is ignored.

In this paper, we study the problem of automatically generating verbal commentaries for sport games. Our main contributions are two-fold. First, to our knowledge we are the first to address this very challenging task. To this end, we built a dataset with 633 pairs of tennis videos and associated commentaries, where each video is annotated with visual attributes such as types of shots, locations of ball bounces, etc. We apply computer vision techniques to automatically detect these attributes. Given the attributes, we propose two machine learning techniques, namely LSTM and structured SVM [9], for commentary generation. Second, we investigate pre-embedding of the tokens and accordingly loss function for LSTM. We propose to use visual similarity between corresponding videos as a proxy for text pre-embeddings based on Canonical Correlation Analysis. Through both objective and human evaluations, we show that with the proposed pre-embedding scheme and loss function, the performance of commentary generation can be substantially improved.

a) Related work: Automatic sports annotation has many potential applications, e.g., content-based video retrieval, object-based video encoding, enhanced broadcast, mobile infotainment, audio description for visually impaired. However, most existing systems only annotate at a crude level, e.g., highlight detection [10], and no attempt is made to generate commentaries for the videos. Tennis is a structured game with a constrained vocabulary which makes it an ideal application for the state-of-the-art methods for generating sentential descriptions. Such commentaries are currently manually generated by tennis enthusiasts worldwide.

In the general field of sentence-based image and video annotation rapid progress has been made in the last few years. The state-of-the-art methods [1], [2], [3], [4], [5], [6] try to “translate” images and videos into language. They typically employ CNNs to encode visual information. RNNs, in particular LSTMs that can keep long term temporal context are used to decode the information into sentences.

The remainder of this paper is structured as follows. We first introduce our new video-commentary dataset in Section II. In Section III we describe the computer vision techniques for detecting visual attributes in tennis, and discuss pre-embedding methods for the tokens in the commentaries. In Section IV we propose a structured SVM and LSTM for generating a commentary from the visual attributes. In particular we incorporate pre-embedded tokens and a novel loss function into the LSTM architecture. In Section V experimental results are presented in terms of retrieval performance and human evaluation. Finally Section VI concludes the paper.

II. A DATASET OF TENNIS VIDEOS AND COMMENTARIES

We collected commentaries for tennis matches from a tennis fan site TennisEarth (http://www.tennisearth.com). The matches are women’s singles from Grand Slam tournaments and involve seven players. Each commentary describes the progress of one point, that is, from a successful service to the point at which the ball goes out of play. We filtered out very brief commentaries such as "Ace!", and corrected errors such as confusion between forehand and backhand. To keep consistency between the matches, we also replaced player names with P1 and P2, where P1 refers to the server of the point. Recognising the two players is in principle possible but it is not in the scope of this paper. Some example commentaries are shown in Table I. Each commentary consists of a variable number of segments separated by commas, where
- Good serve sent at T, P2 returns a backhand return, brief rally, P1 produces a forehand cross-court winner.
- P1 serves an out-wide serve, P2 works a backhand return, P1 comes in quickly and tries a forehand volley but fails to do so.
- Quick serve pointed at T, P2 works a forehand return, P1 produces a forehand cross-court winner.
- P1 goes for a serve in the middle, P2 works a forehand return, couple of shots exchanged, P1 hits a forehand down the line but misses the court.
- Serve aimed at T, P2 backhand return goes wide off the court.

**TABLE I. EXAMPLE COMMENTARIES FOR FIVE GAME POINTS.**

<table>
<thead>
<tr>
<th>match</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td># of points</td>
<td>99</td>
<td>66</td>
<td>105</td>
<td>127</td>
<td>125</td>
<td>111</td>
<td>633</td>
</tr>
<tr>
<td>minutes</td>
<td>15.9</td>
<td>8.4</td>
<td>17.1</td>
<td>21.1</td>
<td>17.3</td>
<td>13.1</td>
<td>92.9</td>
</tr>
</tbody>
</table>

**TABLE II. STATISTICS OF THE DATASET.**

Each segment can be loosely thought of as an independent clause. The segments are of different complexity e.g. the last one typically includes the ID of the player, type of stroke, ball bounce location (and/or direction) as well as winning or losing point.

We also downloaded videos of the matches from YouTube. In total the videos are approximately 12.5 hours in length. We extracted clips corresponding to the collected commentaries. Three example frames from the clips are illustrated in Fig. 1. The resulting dataset contains 633 points, in the form of commentary-video clip pairs, of 92.9 minutes long in total (cf. Table II). Given the complexity of video ground truth annotation in terms of both visual and text features, we consider this dataset as significant. In the rest of this paper, we use match A for learning visual features, matches B and C combined for learning sentence generation, match D for validation, and E and F for testing sentence generation. The dataset will be made available online.

**III. VISUAL AND TEXTUAL FEATURES**

In this section, we describe how to detect visual attributes in tennis videos, and discuss pre-embedding schemes for the commentaries. The focus of the paper is not video processing. However to demonstrate the complexity of the visual analysis and potential sources of noise we summarise the main methods involved in video processing.

**A. Visual features**

With sufficient training examples it may be possible to learn to generate commentaries from pixels. In practice, however, the size of the training set is insufficient for such a learning problem, we therefore drive the generation by key events such as shots and bounces. Commentaries in our dataset tend to focus on the types (forehand or backhand) of the first and last shots in a point, the locations of the first and last bounces, and whether there is a net event. We manually annotated these five attributes for each video, which allows us both to test commentary generation with clean visual attributes, and to learn to detect visual attributes using computer vision techniques.

**b) Forehand or backhand:** For each video, we build a background model using camera motion compensated frames and a median filter [11]. Player tracking is initialised by background subtraction and identifying temporally persistent objects. Kernelised correlation filters [12] are then applied to track the two players independently. HOG3D [13] is employed to describe the spatio-temporal cubes around the players.

In addition we also detect ball candidates by temporal differencing of the frames, and build a ball classifier with shape, colour, and gradient features and an SVM [11]. In each cube around a player, we construct a soft histogram with estimated probabilities of ball candidates falling into spatio-temporal bins. We use the concatenation of HOG3D, ball histogram and a player location flag (top or bottom) as features to train two binary SVMs. The first SVM classifies shot and non-shot, and the second classifies forehand shot and backhand shot.

**c) Location of bounces:** For each video the homographies between frames are estimated, and a mosaic image is built. We detect white lines on the mosaic image using Hough transform. From the set of white lines, vanishing points are found and camera-to-court homography is computed up to an unknown horizontal and vertical scale and unknown position. We then use RANSAC to pair up detected white lines and lines of the tennis court model, and recompute the homography in its entirety. This allows us to detect the tennis court in every frame of the video.

We use the data association technique in [14] to track the tennis ball and detect when it changes its motion. These motion changes are classified into bounces and other types of events using features built from ball trajectories [11]. For the first bounce after serve, we use its relative location with respect to the centre service line and the relevant sideline. For the last bounce, we use a binary flag indicating whether its outside or inside the tennis court.

**d) Detecting net events:** With the detected tennis court and white lines it is straightforward to find the net area. Whether a point contains a net event or not can be distinguished using proximity of the ball to the net and ball trajectory. Finally we augment the five events with the number of frames in the video, forming a 6 dimensional visual feature \( x \) that will drive the commentary generation.

**e) Performance of visual processing:** Visual processing is inevitably prone to errors. With the five visual attributes annotated, we learn their detectors on Match A of the dataset, and report the performance of the detectors on the other five matches in Table III. In the table FH/BH-0 and FH/BH-1...
TABLE III. PERFORMANCE OF VISUAL PROCESSING.

<table>
<thead>
<tr>
<th>FH/BH-0</th>
<th>FH/BH-1</th>
<th>Loc</th>
<th>In/Out</th>
<th>Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.6</td>
<td>22.8</td>
<td>0.064</td>
<td>17.0</td>
<td>18.7</td>
</tr>
</tbody>
</table>

B. Textual features

Most existing techniques for sentence generation treat each word as a token and generate simple sentences without clauses. Tennis commentaries on the other hand contain independent clauses. Given the limited size of the training set, we treat each clause as a token, and generate commentaries as novel combinations of the clauses. This is still a very challenging task as each clause refers to a combination of events that greatly vary within every point as well as between the points. We denote a commentary by a sequence $y = <y_1, \ldots, y_n>$, where $n$ is the number of clauses in the commentary and $y_i$ is the $j$th clause. For example, for the first example commentary in Table I, $n=4$ and $y_1 = \text{Good serve sent at T}$, etc.

The problem with this formulation, however, is that most of the tokens in the training set appear only once, despite the fact that some of them are semantically very similar. e.g. P2 works a forehand return and P2 hits a forehand return. Standard LSTM employs a one-hot representation which ignores the correlation between tokens, and as a result will not be able to learn properly. One solution is to pre-embed the tokens into a semantically meaningful space.

We first consider the bag-of-words representation. We use $z = <z_1, \ldots, z_m>$ to denote the sequence of embeddings of tokens $y = \{y_1, \ldots, y_n\}$, where $z_i \in \mathbb{R}^{d_z}$ are bag-of-words vectors weighted by term frequency-inverse document frequency (TF-IDF), and $d_z$ is the number of unique words in the training set. With this embedding, tokens $P2 \text{ works a forehand return}$ and $P2 \text{ hits a forehand return}$ are close in the feature space.

C. CCA projected features

TF-IDF embedding is not aware of the fact that tokens such as \textit{hugely rally} and \textit{lengthy exchange} between the players are semantically similar. To remedy this, we use the visual modality as a proxy. Given $m$ examples of two sets of variables $\{x_i\}_{i=1}^m$ and $\{z_i\}_{i=1}^m$ where $x_i \in \mathbb{R}^{d_x}$ and $z_i \in \mathbb{R}^{d_z}$, canonical correlation analysis (CCA) finds a projection for each set such that in the joint latent space the linear correlation of the samples is maximised. In our case, $x_i$ is the 6 dimensional visual feature for the $i$th training example, and $z_i = \sum_{j=1}^{n} z_j^*$ is the sum of the TF-IDF embeddings of the tokens for the same example. Once the optimal projections $P_x$ and $P_z$ are found, tokens are embedded from the TF-IDF space into the joint latent space: $z_j^* \leftarrow P_z z_j^*$, where tokens with similar visual groundings are brought close to each other.

IV. LEARNING TO GENERATE COMMENTARIES

In this section, we present a structured SVM and an LSTM for learning to generate commentaries. In particular, we propose to use the Euclidean loss to exploit token correlations in LSTM.

A. Structured SVM

Structured output learning considers problems with structured class labels. It jointly embeds input-output pairs $(x_i, y_i)$ into a feature space and applies linear classifiers. In structured SVM (SSVM), the max-margin hyperplane is sought in the feature space:

$$\min_w \frac{1}{2} ||w||^2 + C \sum_{i=1}^m \xi_i$$

s.t. $w^T \phi(x_i, y_i) - w^T \phi(x_i, y) \geq \Delta(y_i, y) - \xi_i$ for all $y_i \in Y \setminus y_i$, $\xi_i \geq 0$

where $\phi(x, y)$ is the joint embedding, $\Delta(\cdot, \cdot)$ is a label loss function measuring the distance between two labels, $Y$ is the set of all possible labels, and $m$ is the number of training pairs. The prediction of a test example $x$ is given by $y^* = \arg \max_{y \in Y} \omega^T \phi(x, y)$. Iterative cutting-plane algorithms have been developed [9], [15] to solve Eq. (1) in polynomial time.

In our commentary generation task, the labels are sequences $y_i = <y_1, \ldots, y_n>$ with corresponding token embeddings $z_i = <z_1^*, \ldots, z_n^*>$, where $n_i$ is the number of tokens in $y_i$. We define the input-output joint embedding as:

$$\phi(x_i, y_i) = x_i \sum_{j=1}^{n_i} z_j^* ||\text{vec}(\sum_{j=1}^{n_i-1} z_j^* \otimes z_j^{j+1})||$$

where $|| \cdot ||$ denotes vector concatenation, $\otimes$ denotes outer product, and $\text{vec}(-)$ vectorise a matrix. For the label loss function, we use the Euclidean distance between the embeddings of two competing labels: $\Delta(y_i, y) = \sum_{j=1}^{n_i} ||z_j^* - z_j||$.

Finally, it is easy to show that the separation oracle for identifying most violated constraints and for inference is the the Viterbi decoding problem, for which efficient algorithms exist.

B. LSTM

The SSVM in the previous section encodes first-order temporal dependence. In principle SSVM can exploit higher-order dependence but only in a rigid manner. Moreover, learning and inference would be prohibitively expensive. The long short-term memory (LSTM) [8] is a type of recurrent neural network that is capable of learning long temporal dependence while dealing with the vanishing and exploding gradients problem [8].

Fig. 2 illustrates unrolled LSTM model [1]. Our model differs from standard LSTM in the shaded blocks in the figure. First, before applying the token embedder $W_x$, we pre-embed the tokens as $z_i^*$ using either TF-IDF or visually-aware CCA projection, as described in the previous section. Compared to one-hot vectors, such pre-embeddings help exploit correlations between tokens. In Fig. 2, $z_i^*$ corresponds to a special token.
signalling the beginning of each commentary. In practice, we set it to an all-zero vector.

The output of each LSTM, $p(y)$ in Fig. 2, can be interpreted as a set of estimated probabilities for the tokens. Typically the negative log likelihood loss $\ell(y) = -\log p(y)$ or a perplexity based loss are employed for each token. These loss functions again ignore the correlation between the tokens and assume two tokens are either the same or not. In contrast, we propose a loss function based on the pre-embeddings of the tokens:

$$\ell(y) = \frac{1}{2} \sum_{k=1}^{N} p(y(k)) z(k) - z^j \|z^j\|^2$$

(3)

where $j$ as usual is the index of a token in the commentary, $(k)$ is the index of a token in the set of all tokens in the training set, and $N$ is the size of the set.

Defining $e^j = \sum_{k=1}^{N} p(y(k)) z(k)$ as the sum of the embeddings of all tokens weighted by the current estimate of the token probabilities, Eq. (3) can be written as $\ell(y) = \frac{1}{2} \|e^j - z^j\|^2$. It directly follows that

$$\frac{\partial \ell}{\partial e^j} = e^j - z^j \quad \text{and} \quad \frac{\partial \ell}{\partial p(y)} = Z$$

(4)

where $Z = (z^{(1)}, \ldots, z^{(N)})$. Using the chain rule we have

$$\frac{\partial \ell}{\partial p(y)} = Z^T (e^j - z^j) = Z^T \left( \sum_{k=1}^{N} p(y(k)) z(k) - z^j \right)$$

(5)

The gradients computed with Eq. (5) are back-propagated in the network to update the LSTM parameters and the embedders $W_x$ and $W_z$. We use the mean of average loss $1/\ell \sum_{i=1}^{\ell} \left( \frac{1}{n} \sum_{j=1}^{n_i} \ell(y_{ij}) \right)$ on a validation set for model selection and early stopping, where $l$ is the size of the set.

V. EXPERIMENTS

In this section, we evaluate the proposed SSVM and LSTM. We consider the four combinations of visual features (annotated and detected) and token pre-embeddings (TF-IDF and CCA projected), and present both retrieval performance and human evaluation on generated commentaries.

A. Retrieval performance

Metrics commonly used for evaluating sentence generation such as BLEU, ROUGE, METEOR and CIDEr [16] essentially measure how well tokens (words) or n-grams of tokens match. In our application tokens are clauses and almost all tokens differ. Within clauses, existing metrics are also unlikely to capture the semantic similarity between $P1$ backhand fails to

land inside the court and $P1$ over-cooks a backhand in the rally. On the other hand, human judgements are expensive and time-consuming to collect. We reserve human judgements for the four best performing sets of generated commentaries, and report for all runs retrieval performance in terms of recall at rank position $K$ (R@K) and median rank (MR) of the gold item.

For SSVM the affinity between a pair of video and commentary is given by $w^T \phi(x, y)$. For LSTM, we use the Euclidean loss in Eq. (3) as a distance metric between a pair. The results with annotated visual features are shown in Table IV and those with detected visual features are in Table V. The sizes of test sets are 125 (Match E) and 111 (Match F), respectively.

In both tables, CCA projection significantly improves the performance of LSTM. With annotated visual attributes, the median rank drops from 27/18 to 4/3 for Match E and from 30/19 to 4/3 for Match F. With detected visual attributes the improvement is less pronounced, since CCA projections are less accurate in the presence of noise. Nevertheless, the improvement is still significant in many cases, for example, the median rank for video to commentary retrieval reduces from 37 to 24 for Match E, and from 31 to 20 for Match F. On the other hand, with CCA projection the performance of SSVM degrades in most cases. This seems to indicate that the joint embedding in Eq. (2) relies on the sparsity of textual features.

With TF-IDF token pre-embedding, the SSVM performance is comparable with or superior to LSTM. However, with CCA projected pre-embedding, it lags far behind LSTM. As expected, with either learning method and either token pre-embedding scheme, annotated visual features outperform detected ones.

To demonstrate the effectiveness of the proposed pre-embedding scheme and loss function, we also experimented with standard LSTM variants for sentence generation. In the first variant we replace the proposed pre-embeddings with the one-hot version, which assumes independence between the tokens (clauses). In the second variant, we further replace the proposed loss function with the negative log likelihood loss, which is based on $Q$-way softmax where $Q$ is the size of the dictionary. Note that the second version is essentially equivalent to the LSTM in e.g. [1]. Results in Table VI indicate that without the proposed pre-embedding and loss function, the retrieval performance on both test sets degrades to chance performance even with manually annotated visual attributes. This confirms the crucial importance of exploiting token correlations in tennis commentary generation.
TABLE IV. RETRIEVAL PERFORMANCE, ANNOTATED VISUAL FEATURES.

<table>
<thead>
<tr>
<th>Testset: E</th>
<th>Testset: F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video to commentary</td>
<td>Commentary to video</td>
</tr>
<tr>
<td>R@1</td>
<td>R@10</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>SSVM</td>
</tr>
<tr>
<td>LSTM</td>
<td>3.2</td>
</tr>
<tr>
<td>CCA proj.</td>
<td>SSVM</td>
</tr>
<tr>
<td>LSTM</td>
<td><strong>20.0</strong></td>
</tr>
</tbody>
</table>

TABLE V. RETRIEVAL PERFORMANCE, DETECTED VISUAL FEATURES.

<table>
<thead>
<tr>
<th>Testset: E</th>
<th>Testset: F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video to commentary</td>
<td>Commentary to video</td>
</tr>
<tr>
<td>R@1</td>
<td>R@10</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>SSVM</td>
</tr>
<tr>
<td>LSTM</td>
<td>2.4</td>
</tr>
<tr>
<td>CCA proj.</td>
<td>SSVM</td>
</tr>
<tr>
<td>LSTM</td>
<td>2.4</td>
</tr>
</tbody>
</table>

TABLE VI. RETRIEVAL PERFORMANCE WITH BASELINE LSTM, ANNOTATED VISUAL FEATURES.

<table>
<thead>
<tr>
<th>Testset: E</th>
<th>Testset: F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video to commentary</td>
<td>Commentary to video</td>
</tr>
<tr>
<td>R@1</td>
<td>R@10</td>
</tr>
<tr>
<td>LSTM -emb.</td>
<td>-loss</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.8</td>
</tr>
</tbody>
</table>

B. Human evaluation

For each learning method, we employ human evaluation for its best token pre-embedding, that is, TF-IDF for SSVM and CCA projected for LSTM. The annotators are tennis club and national league players, very familiar with tennis commentaries with their specific terminology and able to judge the equivalence between commentaries using different expressions but describing the same semantic events. A generated commentary is compared to the ground truth by scoring individual clauses with 1 or 0 for acceptable or unacceptable description in terms of accuracy, chronological order, ball placement, player identification and types of strokes. The scores are then averaged per commentary and across the whole test data. Examples of the scores are provided in Table VII.

In Table VIII we show the mean and standard deviation of averaged human evaluation scores from three annotators. With annotated visual features, LSTM using the proposed pre-embedding and loss function achieves mean scores of 0.836 and 0.798 on the two test sets, respectively. Note that this scenario is in fact a realistic one. In practice visual features used in this paper can be quickly annotated by a human, by making binary decisions. Natural language commentaries can then be generated using the proposed LSTM. With detected visual features, the best mean scores of 0.733 and 0.689 is also achieved with LSTM. The better performance of LSTM over SSVM confirms its advantage of being able to learn long-term dependences.

C. More details on training

For SSVM the value of the hyper-parameter $C$ is learnt on the validation set from $\{10^{-2}, \ldots, 10^5\}$. For LSTM, we compare the minimal validation loss obtained during training with various settings. Our experiments show that this loss does not vary much with batch size, but it is sensitive to learning rate. We fix the batch size to 50 and learning rate to 0.01 which are optimal values, and plot in Fig. 3 left the minimal validation loss for various combinations of embedding dimensionality in $W_x$ and $W_y$, Columns: dropout ratio. In principle, for a higher embedding dimensionality a stronger regularisation is required. In Fig. 3 right we plot validation loss against...
number of epochs. Each epoch takes less than a second so within minutes LSTM converges, while SSVM typically takes hours.

VI. CONCLUSIONS

We have introduced a new application and proposed an approach for automatic commentary generation for tennis videos relying on a combination of computer vision, natural language processing and machine learning methods. In particular, we address correlations between descriptive terms in the commentary. Both retrieval based evaluation and human evaluation demonstrate that the proposed pre-embedding and loss function for LSTM lead to substantially improved accuracy of the generated commentary over competing methods. We have also introduced a new dataset of 633 annotated pairs of tennis video and corresponding commentary. We demonstrated that the proposed method can learn from a limited number of examples and automatically generate a commentary from a video with over 70% accuracy.

ACKNOWLEDGEMENT

This work has been supported by EU Chist-Era EPSRC EP/K01904X/1 Visual Sense project.

REFERENCES