This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

# Fine-grained Video Captioning for Sports Narrative

Huanyu Yu*, Shuo Cheng*, Bingbing Ni*†, Minsi Wang, Jian Zhang, Xiaokang Yang
Shanghai Institute for Advanced Communication and Data Science,
Shanghai Key Laboratory of Digital Media Processing and Transmission,
Shanghai Jiao Tong University

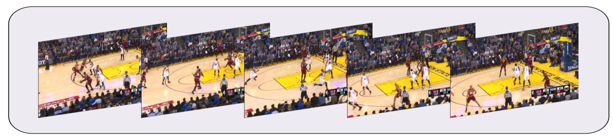
yiranyhy@163.com, acccheng94@gmail.com, nibingbing@sjtu.edu.cn,
mswang1994@gmail.com, stevenash0822@sjtu.edu.cn, xkyang@sjtu.edu.cn

## Abstract

*Despite recent emergence of video caption methods, how to generate fine-grained video descriptions (i.e., long and detailed commentary about individual movements of multiple subjects as well as their frequent interactions) is far from being solved, which however has great applications such as automatic sports narrative. To this end, this work makes the following contributions. First, to facilitate this novel research of fine-grained video caption, we collected a novel dataset called Fine-grained Sports Narrative dataset (FSN) that contains 2K sports videos with ground-truth narratives from YouTube.com. Second, we develop a novel performance evaluation metric named Fine-grained Captioning Evaluation (FCE) to cope with this novel task. Considered as an extension of the widely used METEOR, it measures not only the linguistic performance but also whether the action details and their temporal orders are correctly described. Third, we propose a new framework for fine-grained sports narrative task. This network features three branches: 1) a spatio-temporal entity localization and role discovering sub-network; 2) a fine-grained action modeling sub-network for local skeleton motion description; and 3) a group relationship modeling sub-network to model interactions between players. We further fuse the features and decode them into long narratives by a hierarchically recurrent structure. Extensive experiments on the FSN dataset demonstrates the validity of the proposed framework for fine-grained video caption.*

## 1. Introduction

In spite of recent development of video captioning [37, 38, 42, 18, 30], how to automatically give a fine-grained video description is seldom investigated. One good example of fine-grained video description is Sports Narrative (i.e., especially those team sports such as basketball, soccer, volleyball etc.) Figure 1 shows the difference between conventional video captioning task and fine-grained video description. Note that a caption model can only describe the video from a macroscopic perspective (*e.g.*, a group of people who are playing basketball in the video). In contrast, fine-grained video description is keen on a much more detailed comment about all subjects' individual actions as well as their mutual interactions in the video (*e.g.*, a man passes the ball to his teammate, and his teammate dribbles the ball pass the defenders and gives a slam dunk).



**Ordinary captioning :**
A small group of men are seen running around a basketball court playing a game of basketball.

**Our fine-grained captioning :**
A man passes the ball to his teammate. The ball handler makes a three-point shot but fails. His teammate gets the rebound and makes a slam dunk. The defender tries to block the ball but does not success.

Figure 1: Fine-grained video captioning task versus conventional video captioning task. Fine-grained video caption generates sentences with rich action details and interaction relationships.

For a video involves multiple interacting persons (*e.g.*, team sports), the key task of fine-grained video description is essentially to map multiple spatio-temporal events within the video volume, onto multiple inter-related sentences. In sports video such as basketball game, however, this renders two challenges. First, team sports usually involves a large number of active subjects with complex relationships (*e.g.*, teammates, opponents, defenders) as well as rapid changes of offensive and defensive situation and

*Authors contributed equally to this work.
†Corresponding Author.

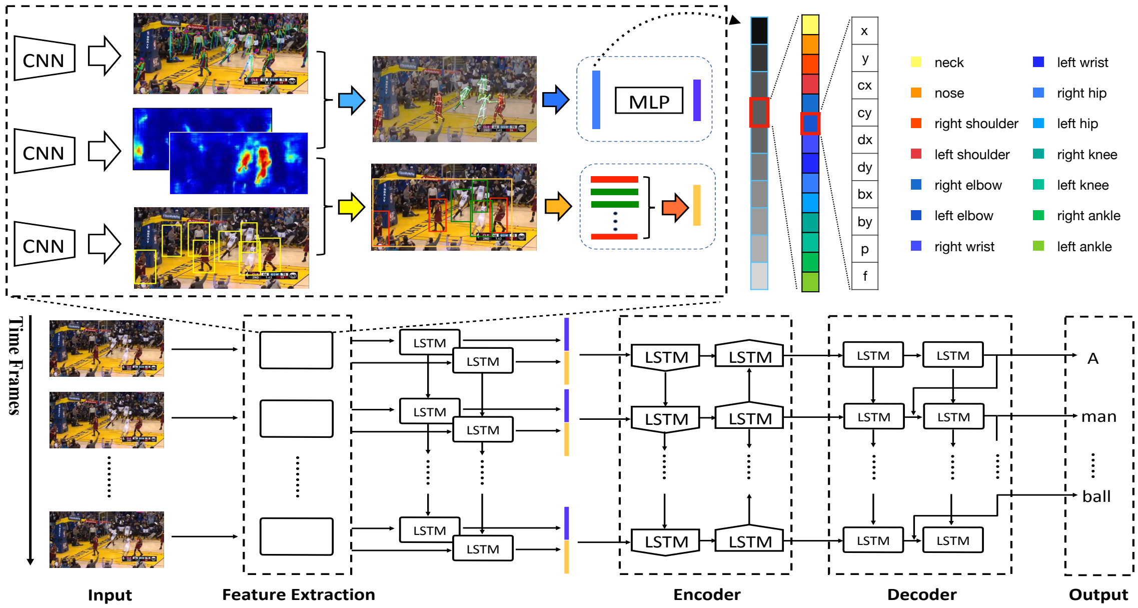



Figure 2: The illustration of our fine-grained video captioning model. The action modeling sub-network utilizes skeletons and optical flows to encode the fine-grained motion details, while the relationship modeling sub-network tackles the interaction analysis among groups. The features from two branch are fused and decoded into narratives via a hierarchically recurrent structure.

attended location, therefore to precisely localize to the important spatio-temporal entities and discover the roles in the activity is difficult. Several recent works attempted to describe the video in multiple sentences [18, 43], however, in these applications video shot boundaries are noticeable (*e.g*., in TACoS-MultiLevel [29], in Acticity Net Captions [18]), making it easy to generate temporal segments, so as to generate description sentences one-by-one accordingly. Also, only one subject is involved [43], which makes it trivial to localize/attend to important movement. Second, important actions/interactions in sports video are very local and fine-grained (i.e., articulated movements), therefore detailed modeling of human macro movements (i.e., trajectories) as well as local skeleton motion and their mutual interactions are required. Unfortunately, most of the previous works [38, 42, 18] only extracts very coarse CNN features for video representation.

To explicitly address these issues and pursue a practical fine-grained sports (i.e., basketball) auto-narrative system, this work introduces a hierarchically grouped recurrent architecture to jointly perform spatio-temporal entities localization and fine-grained motion and interaction modeling. This network features three branches: 1) a spatio-temporal entity localization and role discovering sub-network performs team partition (role discovery) and player localization; 2) a fine-grained action modeling sub-network endowed with an enhanced human skeleton motion description module (i.e., with respect to previous pose recognition techniques [4]) to cope with the task of rapid moving skeleton detection and local motion description; and 3) a group relationship modeling sub-network for modeling interactions among players. We further fuse the features and decode them into narrative languages using a hierarchically recurrent structure.

To kick-off sharable research in this novel area, we introduce a new database, Fine-grained Sports Narrative (FSN), which contains 2,000 NBA basketball HD videos from YouTube website, each of which are annotated with both timestamps and detailed descriptive paragraph. We choose basketball video because basketball video is one of the most challenging videos among all the sports videos, *i.e*., it involves multiple people, interactions of different teams, details of motions, and even outside interference. In the meantime, we propose a novel performance evaluation metric named Fine-grained Captioning Evaluation (FCE), which considers not only the linguistic scores of the sentence (*i.e*., as used by previous coarse-grained video caption tasks) but also whether the key motion and the order of the movement is correctly judged (*i.e*., since these are of great importance in sports video narration). Extensive experiments demonstrate that the proposed novel metric better cope with the fine-grained video captioning task.

## 2. Related Work

Early video captioning methods mainly consider labeling video with metadata [2] clustering videos and describing sentences in order to solve retrieval task. Several previous works [34, 12, 19] generate captions through a language template. Some researchers utilize the recurrent neural networks and LSTM models as sequence decoder on image [39, 15] and video captioning. Later works [38, 9] use CNN features to represent the whole content of the video. [31] detect people in movies to refer to them in their descriptions and to generate correct co-references. Venugopalan *et al*. [37] proposed a new network using a stack of LSTMs to decode the sequence of video frames to generate the corresponding sentence, but all of these works [42, 25, 36, 30] merely focus on single sentence description of the video, which in many cases can not narrate the rich content of the team sports video.

To generate paragraph caption of videos, Yu *et al*. [43] proposed the hierarchical recurrent neural networks (*e.g*., Hierarchical RNN), which consists of sentence generator and paragraph generator. However, it still has some limitations. First, sentences are not located in the videos in the time domain. Second, the generated sentences are highly correlated to the objects occurring in the scene [29]. To tackle the event localization and overlapping problem, dense video captioning is proposed in [18, 32] inspired by the success of dense image captioning [15, 14, 17]. Krishan *et al*. [18] apply DAPs [10] to generate event proposals on the basis of H-RNN [43]. While this work and [3, 11, 23] achieve good results, we notice that the caption of the video is far from detailed (*i.e*., fine-grained). Their model can only describe the video from a macroscopic perspective (*e.g*., A group of people who are playing basketball in the video), and can not describe the detailed movement occurs in the video. We address this problem by proposing a new fine-grained video captioning network and introducing a new dataset FSN, which contains a detailed sports description.

Different from previous methods, which are not appropriate for handling fine-grained video captioning tasks, our method tackles fine-grained action modeling as well as group relationship modeling simultaneously, which enables a new research pipeline for detailed sports video narrative tasks.

# 3. Fine-grained Video Captioning Model

Our goal is to design a fine-grained video captioning module which can narrate the details in sports video with natural language. The main challenges in this task are: first, detect multiple events which may occur simultaneously and localize the discriminative regions on the field; second, recognize the articulate subtle actions of each individual; third, learn complex relationships and complicated interactions among the group of players.

To tackle these problems, we propose a hierarchically grouped recurrent architecture. This network consists of three branch: (1) a spatial-temporal event localization subnetwork generates temporal proposals for event-to-sentence mapping and spatial associative regions for team partition and ball localization; (2) a fine-grained *action modeling sub-network* endowed with an enhanced human skeleton detection module (i.e., with respect to previous pose recognition techniques [4]) to cope with the task of rapid moving skeleton detection and local motion description; (3) an group *relationship modeling sub-network* to model the relationship between players. Finally, we use two LSTM to fuse the features from each branch, and a bi-directional encoder decoder to generate natural language based on the encoded latent feature vectors. We will describe each module in details in the following sections.

## 3.1. Spatial-Temporal Entity Localization and Role Discovering

For fine-grained video captioning task, the first thing is to localize important spatio-temporal entities(*i.e*. the players and balls in the sports game). For localizing important events in a video, we use DAPs [10], an off-the-shell algorithm for accurately detecting events in videos, which provides us with a set of temporal proposals.

Before the model discovers the relationship between players (*i.e*., to generate the caption "A person breaks the opponent's defense and passes the ball to his teammate", the network must form the concept of "**teammate**" and "**defender**"), it is worthwhile localizing important semantic entities, such as ball, team labels of each player. This is similar to previous works on socially aware image/video analysis [27, 8, 7], which solve the problem based on probabilistic graph models. However, their situations only contain simple interactions, the relationship is also defined obscurely, while our task require more accurate partitions.

To achieve this goal, we first pre-train a fully convolutional network to jointly segment out the players and the basketball from the background. Inspired by [24], we use the original cross-entropy loss ($\mathcal{L}_{cross}$) combined with a grouping loss ($\mathcal{L}_{group}$) to optimize the network. Let $\mathbf{P} = \{p_{1,1,1}, ..., p_{H,W,K}\}$ be the output probability map for an input frame and $p_{i,j,k}$ is the predicted probability of class $k$ for pixel $(i, j)$, $H$ and $W$ denote the spatial dimension and $K$ denotes the number of classes (*i.e*., in our case, $K$ = 4 where class 0 indicates background, class 1 and 2 denote the two team, and 3 refers to the ball, respectively). Let $y^*_{i,j} \in \{1, ..., K\}$ be the target class label of pixel $(i, j)$, then the **cross-entropy loss** ($\mathcal{L}_{cross}$) can be write as:

$$\begin{aligned}\mathcal{L}_{cross} = &-\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{1}[y^*_{i,j}=0]\log p_{i,j,0}}{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{1}[y^*_{i,j}=0]} \\ &-\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{1}[y_{y^*_{i,j}}\neq 0]\log(1-p_{i,j,0})}{\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{1}[y_{y^*_{i,j}}\neq 0]},\end{aligned} \tag{1}$$

where $\mathbf{1}[\cdot]$ = 1 iff the condition inside the brackets holds. This cross-entropy loss encourages the network to predict the correct label for each pixel so as to segment out the players from the background. To make the network group the players into two teams, we design another **grouping loss**. Let $\mathbf{S_n} = \{y_1, ..., y_{|S_n|}\}, n \in \{1, 2\}$ be the locations of annotated pixels which are sampled from team $n$, $p_{y_i,c}$ denotes the inferred probability of pixel $y_i$ belonging to team $c$, the grouping loss ($\mathcal{L}_{group}$) is thus defined as following:

$$\mathcal{L}_{group} = \sum_{c=1}^{2}\sum_{n=1}^{2}\frac{1}{|S_n|}\sum_{y_i \in S_n}|p_{y_i,c} - \lfloor\frac{1}{|S_n|}\sum_{y_i \in S_n}p_{y_i,c} + \frac{1}{2}\rfloor| + \sum_{c=1}^{2}\cos|\frac{1}{|S_1|}\sum_{y_i \in S_1}p_{y_i,c} - \frac{1}{|S_2|}\sum_{y_j \in S_2}p_{y_j,c}| \tag{2}$$

the first term minimizes the variance of the predicted classes which should be the same, while the second term maximizes the distance between different classes. The **total loss** ($\mathcal{L}_{total}$) is composed by the weighted sum of both loss. In this network we weight the contribution of the cross-entropy loss with $\lambda_1$ = 1.0 and the grouping loss with $\lambda_2$ = 0.1:

$$\mathcal{L}_{total} = \lambda_1\mathcal{L}_{cross} + \lambda_2\mathcal{L}_{group} \tag{3}$$

By minimizing the total loss, we obtain semantic label for each pixel.

### 3.2. Fine-Grained Action Modeling

To make the generated sentences cover more fine details about individual actions and more diversified, we must provide more useful information (*e.g.*, fine-grained feature), thus we propose an *action modeling sub-network* for exploring individual action information. We observe that action details are highly correlated to player's posture, and the movement of the joints can be used to discern different actions. We use [4] to extract the keypoints for every player.

**Skeleton Grouping** To this end, our task is to encode each extracted skeleton with rich semantic (*e.g.*, team label) and motion information (*e.g.*, optical flow of each joint). In the meantime, we need to remove the irrelevant features such as the skeletons of the audiences. With the probability map computed in Section 3.1, it is easy to group the skeletons into two teams as well as remove the irrelevant background. Let $\mathbf{J}_l = \{(x_{l,1}, y_{l,1}), ..., (x_{l,n}, y_{l,n})\}$ be the detected $l$th skeleton joint set where $(x_{l,i}, y_{l,i})$ is the location of the $i$th joint, $z_l \in \mathbf{Z}$ is the assigned tags for the $l$th skeleton points. Using this notation, we can then formulate our grouping objective as:

$$\mathbf{Z} = \arg\max_z \sum_l \sum_{k=0}^{3}\delta(z_l - k)\sum_i p_{x_{l,i},y_{l,i},k} \tag{4}$$

$\delta[\cdot]$ is the unit pulse function. We obtain the predicted tags for each skeleton by maximizing the assigned probability.

**Skeleton Motion Encoding** We use optical flow to encode the motion of individuals, the objective is to measure in detail the movement of each joint of players. However, as the camera is not fixed, the computed flow fields contain many noises, which are not suitable for representing the movements of the players. To alleviate the influence of relative motion of the camera, we compensate the calculated optical flow field by subtract the estimated camera velocity on $x, y$ direction. To get the corrected optical flow $\mathbf{F'} = \{\mathbf{u'}, \mathbf{v'}\}$, we assume the movement of camera only contains translation for simplifying the problem. Let $\mathbf{F} = \{\mathbf{u}, \mathbf{v}\}$ denotes the original optical flow field calculated by [13], $\mathbf{S} = \{s_1, ..., s_n\}$ denotes the points which belong to the background predicted in Section 3.1, $(\bar{u}, \bar{v})$ denotes the estimated velocity of the camera, then the corrected optical flow $(u'_j, v'_j)$ for every point is re-computed as:

$$\begin{aligned} &\bar{u} = \frac{1}{|\mathbf{S}|}\sum_{s_i \in \mathbf{S}} u_{s_i} \qquad \bar{v} = \frac{1}{|\mathbf{S}|}\sum_{s_i \in \mathbf{S}} v_{s_i} \\ &u'_j = u_j - \bar{u}, \qquad \forall u_j \in \mathbf{u} \\ &v'_j = v_j - \bar{v}, \qquad \forall v_j \in \mathbf{v} \end{aligned} \tag{5}$$

With the computed skeleton joints and their associated motion flow, we design the following individual human skeleton movement descriptor, as inspired by [6]. Formally, the descriptor for every joint contains 10 values, where $x$ and $y$ denotes the position of the joint, $cx$ and $cy$ denotes the arithmetic mean of all the joints belong to the person, $dx$ and $dy$ denotes the optical flow values, bx and by denotes the ball location obtained by the method described in Section 3.1, p is the confidence of the detected joint, f $\in \{-1, 1\}$ is the team flag. The skeleton motion descriptor is depicted in Figure 2. We sort skeletons according to the player's distance from the location of the basketball. For the missing player in the scene, we use zero padding to fix the length of the descriptor. This is analogous to dropout [33], a simple way to resist overfitting.

The underlying design principle of the descriptor is introduced as follows. Center point of the detected skeleton can be used to computed the relative offset of each body part, which makes the posture irrelevant to the location of the player. Optical flow values express the motion of every joint, and thus represent the movement (*e.g.*, velocity and direction) of the player, which is very essential for discerning subtle actions such as "standing" and "walking". Ball location indicates how the nearby player handles the ball (*e.g.*, passes it to teammates or shoots it). In addition, team flag separates the points comes from different teams, which makes the network easier to model the interactions among groups of players.

We reconstruct the features using a multi-layer perceptron network (MLP). The network contains 3 layers with 2048, 1024, 512 hidden units in each layer respectively. To fuse the features across temporal dimension, we use a LSTM (with 1024 hidden units), which demonstrates strong ability to capture long-term dependencies as well as short-term patterns. The LSTM outputs an encoded vector at every step, ready for input to the language generator, which will be introduced in Section 3.3.

### 3.3. Group Relationship Modeling

The action modeling sub-network described above only handles the action details of individual player, but dose not analyze the relationships among players on interactive level. This is insufficient for generating logical sentences that expressing the relationships among players, thus we add another branch called *relationship modeling sub-network* for player interation modeling. Previous works tackle this problem by building graph models [21, 41], or using hierarchy RNNs for high order context modeling [40]. Inspired by [21], we use a simple yet effective way to model the relationship among players.

To analyze the relationship among players, we first localize all the players using [28]. Then the bounding boxes with low confidence according to the probability map in (Section 3.1) are discarded as we only keep 10 of them. To build the scene graph for analyzing the relationships among all the players, we group the 10 atomic bounding boxes (contain only one player) into pairs, and merge them into larger bounding boxes, this will generate 45 extra unique regions.

To obtain the vector representation of each region, we fetch the last stage convolutional feature maps computed by [28], and perform ROI-pooling on the feature maps in each bounding box, the vectors are denoted as $\mathbf{H} = \{h_{i,j}\}, i,j \in \{1, ..., 10\}$, where $h_{i,j} (i = j)$ represent the vector from atomic bounding box only contains one player, and $h_{i,j} (i \neq j)$ represent the vector from merged bounding boxes contain more than one player. As directly concatenating all feature vectors is computational burdensome, we merge the feature vectors by a gate function, which determines the weight for each vector. The merged feature vector $\bar{h}$ is computed as follows:

$$\bar{h} = \frac{1}{\|\mathbf{H}\|} \sum_{i=1}^{10} \left( \sum_{j \neq i} \sigma_{\langle i,j \rangle} \left( h_{i,i}, h_{i,j} \right) h_{i,j} + h_{i,i} \right) \tag{6}$$

$\sigma_{\langle i,j \rangle}$ denotes the gate function, which can be unrolled as:

$$\sigma_{\langle i,j \rangle} \left( h_{i,i}, h_{i,j} \right) = \textbf{sigmoid} \left( \omega \cdot [h_{i,i}, h_{i,j}] \right), \tag{7}$$

where $[\cdot]$ denotes concatenating, $\omega$ denotes the transformation matrix, which can be optimized using standard back-propagation algorithm. The designed gate function learns to assign weight for different merged regions according to the interaction pattern inside bounding box, and controls how much the region contributes to the final averaged feature vector. If the region does not contain any interactions or the interactive relationship is not required to be modeling during captioning, the gate function will out a low value, reduce its effect for subsequent feature calculating.

To pass the useful interactive information along temporal dimension, we use a LSTM with 1024 hidden units. This LSTM do not share weight with the LSTM in Section 3.2, as the features comes from different levels with different granularities (*e.g.*, the skeletons depict the articulated action details, while the interaction features contain more about group relationships).

### 3.4. Narrative Generation

Once we obtain the individual action feature vectors and relationship feature vectors by above methods, the next stage of our pipeline is to generate natural language description. Different from sentence generation, paragraph generation must take care of the relative contexts and the relationships between generated sentences.

The natural language generation module of our pipeline uses an encode-decoder architecture. The encoder is a two-layer bi-directional LSTM, which fuses the action features and relationship features cross all frames in a video and encodes them into a latent space. The decoder contain a sentence LSTM and a paragraph LSTM (*i.e.*, the former generates current word according to the sentence state while the latter provides semantic context about previous generated sentences), See Figure 2 for illustration. During decoding, the decoder decodes the latent vector and reasons about the current word according to sentence context cues and paragraph context cues. The decoder outputs a distribution $\mathbf{P}$ about all words in vocabulary set at every time step:

$$\mathbf{P} \left( w_t^n | c_{1:n-1}, w_{t-1}^n, h_{t-1} \right), \tag{8}$$

where $h_{t-1}$ denotes the hidden state from time step $t-1$, $c_{1:n-1}$ denote the output of the paragraph LSTM, $w_t^n$ is the $t$th word in sentence $n$, respectively. We train the language generation module by minimizing the caption loss ($\mathcal{L}_{cap}$), which is defined as:

$$\mathcal{L}_{cap} = -\sum_{n=1}^{N} \sum_{t=1}^{T_n} \log \mathbf{P} \left( w_t^n | c_{1:n-1}, w_{t-1}^n, h_{t-1} \right) \Big/ \sum_{n=1}^{N} T_n, \tag{9}$$

where $T_n$ is the number of words in the sentence $n$.

### 3.5. Training and Optimization

For training the segmentation model in Section 3.1, we initialize the model using a Gaussian distribution with standard deviation of 0.05. Then the model is optimized by Stochastic Gradient Descent (SGD) algorithm, with batch size of 8. We set momentum to 0.9, and weight decay to 0.0005. The initial learning rate is set to 0.0016 and we linearly reduce it to 0 in the following 100 epochs.

For training the action modeling sub-network, the relationship modeling sub-network and the language generation module, we initialize the model using a Gaussian distribution with standard deviation of 0.01, the batch size is reduced to 1. The initial learning rate is 0.001 and we use Adam [16] and use default configurations to optimize it in the following 300 epochs.We train our models on two GTX TITAN X, it takes about 70 hours for the model to converge.

# 4. Fine-grained Sports Narrative Dataset and New Evaluation Metrics

Fine-grained Sports Narrative Dataset (FSN) is a mutiperson sports video captioning dataset. Each video is annotated with a paragraph of detailed description consisting of several sentences. Distinguished from the previous video captioning datasets, which all describe the motion from a macro perspective, this dataset focuses more on the detailed motion of the subjects. Each sentence covers an unique segment of the video. We allow the segments to overlap in time domain. Next, we introduce the collection process of the dataset and present detailed statistical analysis on this dataset. After these, we gives a detailed description of the new evaluation metric FCE.

## 4.1. Dataset collection

We collect 50 original NBA HD game video on Youtube website and split them to 6000 segments. We then remove the videos that are too short and of poor visual quality and select 2000 videos with detailed and diverse motions as the final annotation videos. All the videos are of high quality and have audio channel. Our annotation task includes two steps. First, we make a description of the events occur in the video according to the way used in basketball commentary that each sentence describes one movement of the moving subject. Second, we mark the starting and ending times of each statement described. Since the players in basketball videos always move very quickly, we use a dedicated annotation tool to slow down the speed five times (*i.e.*, 1/30 s per frame) to ensure annotation accuracy.

## 4.2. Dataset statistics

Our dataset contains 2K videos, with an average of 3.16 labeled sentences per video, for a total of 6520 sentences. Each video has an average of 29.7 description words. On average, each sentence describes 1.8s in video and 29.7% of the entire video. The whole paragraph for each video on average describes the 93.8% of the entire video, which demonstrate that our annotations basically covers the main events in the video.

We make a parts of speech analysis on our dataset compared with ActivityNet Captions. As is shown in Figure 3, the FSN dataset has more verbs, which demonstrate this fine-grained dataset pay more attention to the motion of the subject. In Table 1, the comparison of our dataset with MSR-VTT, M-VAD and ActivityNet Captions further demonstrates the **fine-grained details** of our captioning annotations. FSN dataset has the most sentences per second of 0.556, while the other dataset are all below 0.1, this reflects that our dataset are of more detailed descriptions. Furthermore, we find that our dataset has 1.67 verbs in a sentence on average, comparing to 1.41 for ActivityNet Captioning and 1.37 for MSR-VTT respectively. Similarly, the verb ration of our dataset which is computed by dividing verbs-per-sentence by words-per-sentence is also much higher than other three datasets. This demonstrates that our dataset pay more attention to the motion of the subjects, which is consistent with our objective, *i.e.*, fine-grained video description.

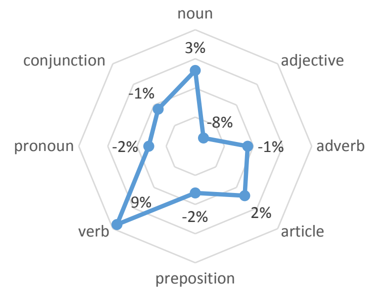



Figure 3: The parts of speech distribution of FSN dataset compared with ActivityNet Captions. All the values in this figure are the differences between these two datasets in the percentage form. There are more verbs in FSN dataset, as this is a fine-grained captioning dataset focusing more on detailed actions.

| Dataset | Sentences per Second | Verbs per Sentence | Verb Ratio |
|---|---|---|---|
| MSR-VTT | 0.067 | 1.37 | 14.8% |
| YouCook | 0.056 | 1.33 | 12.5% |
| ActivityNet Captions | 0.028 | 1.41 | 10.4% |
| FSN (ours) | **0.556** | **1.67** | **18.3%** |

Table 1: Comparisons of different video caption datasets.

## 4.3. Evaluation Metrics

Observing the fact that previous metrics **can not** evaluate the captions of fine-grained sports video appropriately, we introduce **Fine-grained Captioning Evaluation** (FCE) metric. To focus on motions and their temporal order, we compute a motion penalty for a given pair of the candidate sentence and the reference sentence. We label all the verbs or derivation of verbs in the training dictionary and identify all the word by $(c_v, c_n)$ in the candidate sentences and $(r_v, r_n)$ in the reference sentences, where we use $v, n$ to denote verbs and non-verbs respectively. We match the unigrams by the same mapping criterion used in [20]. We use $m_i(c_v)$ to represent the number of the verb unigrams that is covered in each matcher $m_i$ and $n_{cv}$ for the total number of the verbs in this translation. First, the verb precision is computed as the ratio of the number of the verb unigrams in the candidate sentence that is mapped to the total number of the verb unigrams:

$$P_{v-m} = \frac{\sum_i m_i(c_v)}{n_{cv}} \tag{10}$$

Second, we compute the order precision which penalize the score if the order of the verb is incorrect. We consider a wrong order has occurred if and only if the following formula is evaluates to a negative number:

$$[p(m_i(c_v)) - p(m_j(c_v))] \cdot [p(m_i(r_v)) - p(m_j(r_v))], \tag{11}$$

where $p(m_i(c_v))$ denote the position of the matched unigram $m_i(c_v)$ in the candidate sentences and $p(m_i(r_v))$ denote the position of the matched unigram $m_i(r_v)$ in the reference sentences. When the resulting value of the above formula is negative, we assign $E_{i,j}$ to 1. Then we sum all the $E_{i,j}$ to get the total number of the order error. We divide the order error by $\binom{2}{m(c_v)}$ to get the ratio of order error and then we use its complement to denote the order accuracy. The final accuracy of the verb consists of the verb precision and the order accuracy:

$$P_{v-acc} = \left(\frac{\sum_i m_i(c_v)}{\sum c_v}\right)^{1/2} \cdot \left(1 - \frac{\sum_{i=1}^{m(c_v)} \sum_{j=1}^{m(c_v)} E_{i,j}}{\binom{2}{m(c_v)}}\right)^{1/2}$$

$$E_{i,j} = \begin{cases} 1 & if\ [p(m_i(c_v)) - p(m_j(c_v))] \\ & \cdot [p(m_i(r_v)) - p(m_j(r_v))] < 0 \\ 0 & otherwise \end{cases} \tag{12}$$

We calculate the linguistic score $F_{lin}$ of the captioning sentence using the method in METEOR since it has shown better correlation with human subjects. Finally, the FCE Score for the given sentence is computed as follows:

$$Score = F_{lin} \cdot P_{v-acc} \tag{13}$$

We report scores of FCE and other traditional evaluation metrics such as Bleu, METEOR, Rouge-L and CIDEr-D in the followings. We also conduct a comparison between FCE and other traditional metrics. More details can be viewed in our supplementaries. We will release our dataset as well as the evaluation tools.

# 5. Experiments

In this section we first evaluate our model on its ability of generating fine-detailed descriptions. We conduct experiments on FSN dataset, which is built specifically for this task. Next, we analyze each component of our full model and, this ablation study is very useful for identifying the effect of each module in our whole pipeline, and find out the most important part for improving fine-grained video captioning tasks.

## 5.1. Captioning Results

To evaluate the generated results, we first employ four different traditional metrics: Bleu (B) [26], METEOR (M) [20], CIDEr-D (C) [35], SPICE (S) [1] and Rouge-L (R) [22], we calculate the metrics using COCO evaluation tools [5]. We compare our full model with some state of art methods on traditional video captioning task: S2VT [37], LSTM-YT [38], H-RNN [43] and DenseCap-event [18]. The quantitative results are illustrated in Table 2, FCE is short as F. Human evaluation is also used to make the result more convincing and the evaluation details are in Supplementary.

We find LSTM-YT performs worse than other models as it encodes whole video sequences into vectors by mean pooling. This will loss important information which are necessary for discerning articulate actions. We notice although H-RNN and DenseCap-event are able to generate fluent sentences as they take context into account, the generated sentences contain inaccurate action details of the players. Different from previous methods, our model generates more detailed sentences, which accurately describe fine grained actions of the player and interactions among the group.

In addition, we also measure the generated results with the introduced FCE metric. Comparing to METEOR, we find a variance drop on scores among all the method (marked with blue). While LSTM-YT drops the most $\left(\frac{0.1304-0.2211}{0.2211} = -41\%\right)$, our full model drops less severe than other method $\left(\frac{0.1944-0.2757}{0.2757} = -29\%\right)$ as it is able to generate more accurate action details, *i.e.*, the new metric focus more on fine-grained actions.

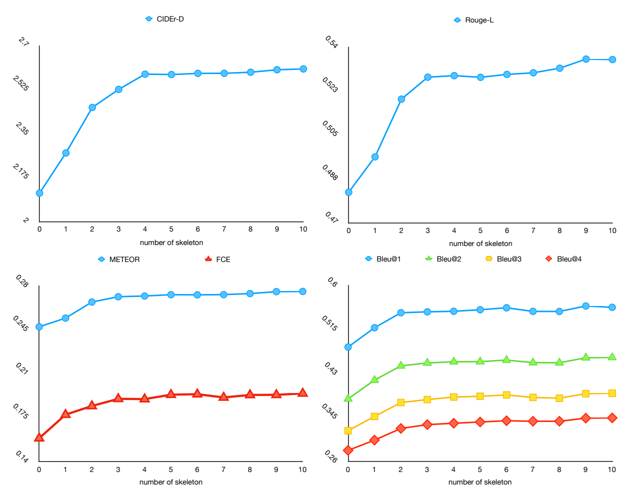



Figure 4: Visualizations of the relationships between the number of used skeletons and the evaluation metrics. Best viewed in colors.

## 5.2. Ablation Study

To analyze the effect of the incorporated skeletons, we conduct detailed experiments on the action modeling branch. We evaluate the captioning results of the model trained with different number of skeletons, see Figure 4 for more details. We find that utilizing skeleton features can greatly improve the caption results as it provides the model with more fine-detailed information. In addition, the first few skeletons contribute the most improvement. This is reasonable because most of our ground-truth paragraphs describe 2-3 players, which is in line with the actual narrative situation.

In addition, we also measure the effort of using optical flow

| | C | B@4 | B@3 | B@2 | B@1 | R | M | S | F | Human |
|---|---|---|---|---|---|---|---|---|---|---|
| LSTM-YT [38] | 1.88 | 0.2663 | 0.2891 | 0.3512 | 0.4551 | 0.4508 | 0.2211 | 0.331 | 0.1304 (-41%) | 3.207 |
| S2VT [37] | 2.10 | 0.2804 | 0.3101 | 0.3712 | 0.4762 | 0.4729 | 0.2394 | 0.346 | 0.1512 (-38%) | 3.536 |
| H-RNN [43] | 2.09 | 0.2767 | 0.3043 | 0.3671 | 0.4632 | 0.4661 | 0.2331 | 0.342 | 0.1508 (-34%) | 3.374 |
| DenseCap-event [18] | 2.23 | 0.2962 | 0.3327 | 0.3997 | 0.4912 | 0.4893 | 0.2522 | 0.358 | 0.1617 (-36%) | 3.913 |
| only relation (ours) | 2.11 | 0.2817 | 0.3197 | 0.3812 | 0.4812 | 0.4822 | 0.2475 | 0.351 | 0.1587 (-36%) | 3.211 |
| only action (ours) | 2.28 | 0.3070 | 0.3518 | 0.4200 | 0.5180 | 0.4933 | 0.2589 | 0.363 | 0.1708 (-35%) | 3.854 |
| full model (ours) | **2.61** | **0.3445** | **0.3921** | **0.4612** | **0.5580** | **0.5350** | **0.2757** | **0.391** | **0.1944** (-29%) | **4.224** |

Table 2: We report our CIDEr-D (C), METEOR (M), Bleu (B), Rouge-L (R), SPICE(S) and FCE (F) scores comparing with other state-of-the-art methods. The drop percentage using FCE comparing with METEOR is marked in the brackets.


Figure 5: Comparison of paragraphs generated by our full model with its downgraded versions (*e.g.*, without optical flow or team flag).

(short as OF) as well as team flag (short as TF). We find aligned optical flow provides the model with more accurate motion informations, which are necessary for discerning articulated subtle actions, while the team flag helps the network to distinguish defenders and the teammates. See Figure 5 for more qualitative results.

## 6. Conclusions

In this paper we propose the Fine-grained Sports Narrative Dataset for fine-grained video captioning task. Observing the fact that conventional evaluation metrics are not appropriate for evaluating the performance, we introduce a metric named Fine-grained Captioning Evaluation (FCE). To benchmark the dataset, we report the performance of our method.

## 7. Acknowledgement

This work was supported by National Science Foundation of China (U161146161502301,61671298,61521062). The work was partially supported by Chinas Thousand Youth Talents Plan State Key Research and Development Program (2016YFB1001003), and 18DZ2270700.

## References


[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. SPICE: semantic propositional image caption evaluation. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, pages 382–398, 2016. 7

[2] H. B. Aradhye, G. Toderici, and J. Yagnik. Video2text: Learning to annotate video content. In *ICDM Workshops 2009, IEEE International Conference on Data Mining Workshops, Miami, Florida, USA, 6 December 2009*, pages 144–151, 2009. 2

[3] L. Baraldi, C. Grana, and R. Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3185–3194, 2017. 3

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016. 2, 3, 4

[5] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollr, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv:1504.00325*, 2015. 7

[6] G. Chéron, I. Laptev, and C. Schmid. P-CNN: pose-based CNN features for action recognition. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3218–3226, 2015. 4

[7] L. Ding and A. Yilmaz. Learning relations among movie characters: A social network perspective. In *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, pages 410–423, 2010. 3

[8] L. Ding and A. Yilmaz. Inferring social relations from visual concepts. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 699–706, 2011. 3

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2625–2634, 2015. 3

[10] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, pages 768–784, 2016. 3

[11] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1141–1150, 2017. 3

[12] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. J. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 2712–2719, 2013. 2

[13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 4

[14] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4565–4574, 2016. 3

[15] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137, 2015. 3

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. abs/1412.6980, 2014. 6

[17] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. *CoRR*, abs/1611.06607, 2016. 3

[18] R. Krishna, K. Hata, F. Ren, F. Li, and J. C. Niebles. Dense-captioning events in videos. *CoRR*, abs/1705.00754, 2017. 1, 2, 3, 7, 8

[19] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013. 2

[20] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, pages 65–72, 2005. 6, 7

[21] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and caption regions. *CoRR*, abs/1707.09700, 2017. 5

[22] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004. 7

[23] X. Long, C. Gan, and G. de Melo. Video captioning with multi-faceted attention. *CoRR*, abs/1612.00234, 2016. 3

[24] A. Newell and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. *CoRR*, abs/1611.05424, 2016. 3

[25] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4594–4602, 2016. 3

[26] K. Papineni, S. Roukos, T. Ward, and W. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 311–318, 2002. 7

[27] V. Ramanathan, B. Yao, and F. Li. Social role discovery in human events. In *2013 IEEE Conference on Computer Vision

*and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2475–2482, 2013. 3

[28] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 5

[29] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition - 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings*, pages 184–195, 2014. 2, 3

[30] A. Rohrbach, M. Rohrbach, and B. Schiele. The long-short story of movie description. In *Pattern Recognition - 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*, pages 209–221, 2015. 1, 3

[31] A. Rohrbach, M. Rohrbach, S. Tang, S. J. Oh, and B. Schiele. Generating descriptions with grounded and co-referenced people. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4196–4206, 2017. 3

[32] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y. Jiang, and X. Xue. Weakly supervised dense video captioning. *CoRR*, abs/1704.01502, 2017. 3

[33] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 4

[34] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. J. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1218–1227, 2014. 2

[35] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575, 2015. 7

[36] S. Venugopalan, L. A. Hendricks, R. J. Mooney, and K. Saenko. Improving lstm-based video description with linguistic knowledge mined from text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1961–1966, 2016. 3

[37] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4534–4542, 2015. 1, 3, 7, 8

[38] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1494–1504, 2015. 1, 2, 3, 7, 8

[39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164, 2015. 3

[40] M. Wang, B. Ni, and X. Yang. Recurrent modeling of interaction context for collective activity recognition. 5

[41] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. *CoRR*, abs/1701.02426, 2017. 5

[42] L. Yao, A. Torabi, K. Cho, N. Ballas, C. J. Pal, H. Larochelle, and A. C. Courville. Describing videos by exploiting temporal structure. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 4507–4515, 2015. 1, 2, 3

[43] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4584–4593, 2016. 2, 3, 7, 8