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).

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
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 ActivityNet 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 sub-network 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 (Lcross) combined with a grouping loss (Lgroup) to optimize the network. Let \( \mathbf{P} = \{p_{i,j,1}, ..., p_{H,W,K}\} \) 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 \( \mathcal{L}_{\text{cross}} \) can be write as:

\[
\mathcal{L}_{\text{cross}} = -\frac{\sum_{i=1}^{H} \sum_{j=1}^{W} 1[y_{i,j}^* = 0] \log p_{i,j,0}}{\sum_{i=1}^{H} \sum_{j=1}^{W} 1[y_{i,j}^* = 0]} - \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} 1[y_{i,j}^* \neq 0] \log (1 - p_{i,j,0})}{\sum_{i=1}^{H} \sum_{j=1}^{W} 1[y_{i,j}^* \neq 0]},
\]

where \( 1[.] = 1 \) 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 $S_n = \{y_1, ..., y_{|S_n|}\}$, $n \in \{1, 2\}$ 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 ($\mathcal{L}_{\text{group}}$) is thus defined as following:

$$\mathcal{L}_{\text{group}} = \sum_{c=1}^{2} \sum_{n=1}^{2} \frac{1}{|S_n|} \sum_{y_i \in S_n} \left| p_{y_i,c} - \frac{1}{|S_n|} \sum_{y_j \in S_n} p_{y_j,c} + \frac{1}{2} \right|$$

$$\quad + \sum_{c=1}^{2} \cos \frac{1}{|S_1|} \sum_{y_i \in S_1} p_{y_i,c} - \frac{1}{|S_2|} \sum_{y_i \in S_2} p_{y_i,c}$$

(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 ($\mathcal{L}_{\text{total}}$) 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 $\lambda_2 = 0.1$:

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

(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 skeleton group computed in Section 3.1 as the skeletons of the audiences. With the probability map meantime, we need to remove the irrelevant features such as skeleton joint set where $z_i$ joint, we design another grouping loss $c_n$ 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 ($\mathcal{L}_{\text{group}}$) is thus defined as following:

$$\mathcal{L}_{\text{group}} = \sum_{c=1}^{2} \sum_{n=1}^{2} \frac{1}{|S_n|} \sum_{y_i \in S_n} \left| p_{y_i,c} - \frac{1}{|S_n|} \sum_{y_j \in S_n} p_{y_j,c} + \frac{1}{2} \right|$$

$$\quad + \sum_{c=1}^{2} \cos \frac{1}{|S_1|} \sum_{y_i \in S_1} p_{y_i,c} - \frac{1}{|S_2|} \sum_{y_i \in S_2} p_{y_i,c}$$

(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 ($\mathcal{L}_{\text{total}}$) 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 $\lambda_2 = 0.1$:

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

(3)

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

**Skeleton Grouping** To this end, our task is to encode each skeleton group computed in Section 3.1 as the skeletons of the audiences. With the probability map meantime, we need to remove the irrelevant features such as skeleton joint set where $z_i$ joint, we design another grouping loss $c_n$ 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 ($\mathcal{L}_{\text{group}}$) is thus defined as following:

$$\mathcal{L}_{\text{group}} = \sum_{c=1}^{2} \sum_{n=1}^{2} \frac{1}{|S_n|} \sum_{y_i \in S_n} \left| p_{y_i,c} - \frac{1}{|S_n|} \sum_{y_j \in S_n} p_{y_j,c} + \frac{1}{2} \right|$$

$$\quad + \sum_{c=1}^{2} \cos \frac{1}{|S_1|} \sum_{y_i \in S_1} p_{y_i,c} - \frac{1}{|S_2|} \sum_{y_i \in S_2} p_{y_i,c}$$

(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 ($\mathcal{L}_{\text{total}}$) 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 $\lambda_2 = 0.1$:

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

(3)

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

**Skeleton Grouping** To this end, our task is to encode each skeleton group computed in Section 3.1 as the skeletons of the audiences. With the probability map meantime, we need to remove the irrelevant features such as skeleton joint set where $z_i$ joint, we design another grouping loss $c_n$ 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 ($\mathcal{L}_{\text{group}}$) is thus defined as following:

$$\mathcal{L}_{\text{group}} = \sum_{c=1}^{2} \sum_{n=1}^{2} \frac{1}{|S_n|} \sum_{y_i \in S_n} \left| p_{y_i,c} - \frac{1}{|S_n|} \sum_{y_j \in S_n} p_{y_j,c} + \frac{1}{2} \right|$$

$$\quad + \sum_{c=1}^{2} \cos \frac{1}{|S_1|} \sum_{y_i \in S_1} p_{y_i,c} - \frac{1}{|S_2|} \sum_{y_i \in S_2} p_{y_i,c}$$

(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 ($\mathcal{L}_{\text{total}}$) 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 $\lambda_2 = 0.1$:

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

(3)

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

**Skeleton Grouping** To this end, our task is to encode each skeleton group computed in Section 3.1 as the skeletons of the audiences. With the probability map meantime, we need to remove the irrelevant features such as skeleton joint set where $z_i$ joint, we design another grouping loss $c_n$ 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 ($\mathcal{L}_{\text{group}}$) is thus defined as following:

$$\mathcal{L}_{\text{group}} = \sum_{c=1}^{2} \sum_{n=1}^{2} \frac{1}{|S_n|} \sum_{y_i \in S_n} \left| p_{y_i,c} - \frac{1}{|S_n|} \sum_{y_j \in S_n} p_{y_j,c} + \frac{1}{2} \right|$$

$$\quad + \sum_{c=1}^{2} \cos \frac{1}{|S_1|} \sum_{y_i \in S_1} p_{y_i,c} - \frac{1}{|S_2|} \sum_{y_i \in S_2} p_{y_i,c}$$

(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 ($\mathcal{L}_{\text{total}}$) 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 $\lambda_2 = 0.1$:

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

(3)

By minimizing the total loss, we obtain semantic label for each pixel.
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 does not analyze the relationships among players on interactive level. This is insufficient for generating logical sentences that express the relationships among players, thus we add another branch called relationship modeling sub-network for player interaction 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 $H = \{h_{i,j}\}, i, j \in \{1, ..., 10\}$, 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:

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

where $\sigma_{(i,j)}$ denotes the gate function, which can be unrolled as:

$$\sigma_{(i,j)}(h_{i,i}, h_{i,j}) = \text{sigmoid}(\omega \cdot [h_{i,i}, h_{i,j}]),$$

where $[,]$ denotes concatenating, $\omega$ denotes the transformation matrix, which can be optimized using standard backpropagation 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 output 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 $P$ about all words in vocabulary set at every time step:

$$P (w_{t}^{n} | c_{1:n−1}^{n}, w_{t−1}^{n}, h_{t−1})$$,

where $h_{t−1}$ denotes the hidden state from time step $t − 1$, $c_{1:n−1}^{n}$ 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 ($L_{cap}$), which is defined as:

$$\mathcal{L}_{cap} = - \sum_{n=1}^{N} \sum_{t=1}^{T_{n}} \log P (w_{t}^{n} | c_{1:n−1}^{n}, w_{t−1}^{n}, h_{t−1}) \bigg/ \sum_{n=1}^{N} T_{n},$$

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 multi-person 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 1.8 s 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 ratio 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Sentences per Second</th>
<th>Verbs per Sentence</th>
<th>Verb Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR-VTT</td>
<td>0.067</td>
<td>1.37</td>
<td>14.8%</td>
</tr>
<tr>
<td>YouCook</td>
<td>0.056</td>
<td>1.33</td>
<td>12.5%</td>
</tr>
<tr>
<td>ActivityNet Captions</td>
<td>0.028</td>
<td>1.41</td>
<td>10.4%</td>
</tr>
<tr>
<td>FSN (ours)</td>
<td>0.556</td>
<td>1.67</td>
<td>18.3%</td>
</tr>
</tbody>
</table>

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 \( m_v(c_v) \) to represent the number of the verb unigrams that is covered in each matcher \( m_v \) and \( n_v \) 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:

$$P_{v-m} = \frac{\sum_i m_i(c_v)}{\#c_v}$$

(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:

$$[p(m_i(c_v)) - p(m_j(c_v))] \cdot [p(m_i(r_v)) - p(m_j(r_v))]$$

(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 \(\sum m(c_v)\) 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:

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

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

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:

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

(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 \((-29%)\), our full model drops less severe than other method \((-14%)\) 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.](image)

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
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>C</th>
<th>B@4</th>
<th>B@3</th>
<th>B@2</th>
<th>B@1</th>
<th>R</th>
<th>M</th>
<th>S</th>
<th>F (%)</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-YT [38]</td>
<td>1.88</td>
<td>0.2663</td>
<td>0.2891</td>
<td>0.3512</td>
<td>0.4551</td>
<td>0.4508</td>
<td>0.2211</td>
<td>0.331</td>
<td>0.1304 (-41%)</td>
<td>3.207</td>
</tr>
<tr>
<td>S2VT [37]</td>
<td>2.10</td>
<td>0.2804</td>
<td>0.3101</td>
<td>0.3712</td>
<td>0.4762</td>
<td>0.4729</td>
<td>0.2394</td>
<td>0.346</td>
<td>0.1512 (-38%)</td>
<td>3.536</td>
</tr>
<tr>
<td>H-RNN [43]</td>
<td>2.09</td>
<td>0.2767</td>
<td>0.3043</td>
<td>0.3671</td>
<td>0.4632</td>
<td>0.4661</td>
<td>0.2331</td>
<td>0.342</td>
<td>0.1508 (-34%)</td>
<td>3.374</td>
</tr>
<tr>
<td>DenseCap-event [18]</td>
<td>2.23</td>
<td>0.2962</td>
<td>0.3327</td>
<td>0.3997</td>
<td>0.4912</td>
<td>0.4893</td>
<td>0.2522</td>
<td>0.358</td>
<td>0.1617 (-36%)</td>
<td>3.913</td>
</tr>
</tbody>
</table>

**only relation (ours)**
- 2.11 0.2817 0.3197 0.3812 0.4812 0.4822 0.2475 0.351 43
- 2.23 0.2962 0.3327 0.3997 0.4912 0.4893 0.2522 0.358 35
- 2.61 0.3445 0.3921 0.4612 0.5580 0.5350 0.2757 0.391 18
- 4.224 34

**only action (ours)**
- 0.1708 (-37)
- 0.1508 (-36)
- 0.1944 (-36)
- 0.1304 (-36)

**full model (ours)**
- 2.61 0.3445 0.3921 0.4612 0.5580 0.5350 0.2757 0.391 0.1944 (-29%) 4.224

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


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


