Automated Basketball Video Captioning

Beri Kohen Behar
ICME
Stanford University
bkohen@stanford.edu

Lucas Pauker
Department of Computer Science
Stanford University
lpauker@stanford.edu

Abstract

In this paper, we investigate automated machine captioning of basketball videos. We expand on the paper Sports Video Analysis on Large Scale Data [1], which provides a novel NBA video captioning dataset, and a unified approach for video captioning, the SportsFormer model. We identify a main shortcoming of the paper's approach, which is that it uses a very complex pipeline that involves basketball specific vision models. We demonstrate that with additional data gathering, modifications to the decoder architecture and a multi-task learning framework, we can have a simpler and high performing model composed of a pre-trained video transformer, a pre-trained encoder to process the video encodings, and a decoder model. This approach eliminates the use of three object detection models, a segmentation model, a vision transformer, and two other transformers from the SportsFormer architecture. Our method removes the need for training basketball specific vision models, which potentially makes this framework more adaptable to other sports.

1 Key Information to Include

- External collaborators (if you have any): None.
- Mentor (custom project only): Yuan Gao.
- Sharing project: Lucas is using this codebase for CS324 as well.

2 Introduction

In this paper, we investigate automated captioning of basketball videos. Specifically, we develop a simple model that takes a basketball video as input and generates play-by-play analysis automatically in the form of text.

This problem is interesting for two reasons. First, it has potential applications in professional sports analysis and broadcasting. By automatically generating commentary, our proposed model can help sports analysts to remember the game better, perform statistical analysis, and easily retrieve key plays. Second, this problem is technically challenging because captioning sports videos requires understanding multiple aspects of the game, which can be difficult due to the many actions that can happen in a single play. Finally, another technical challenge with video captioning (compared to standard natural language generation) is that processing videos require significantly more storage and computation.

Although some previous work has been done on this problem [1, 2], it typically involves complex basketball-specific modeling approaches such as ball detection and player skeleton detection. In contrast, we aim to develop a simpler, sport-agnostic model.

We base our work on Sports Video Analysis on Large Scale Data [1], which provides a novel NBA video captioning dataset, and a unified approach for video captioning, the SportsFormer model. We experiment with two new approaches. The first is focused on using additional data to restrict the caption generation model to use only player names that play for one of the two teams on court. The second approach is focused on using additional labels provided in our dataset (action and player labels) to train a model with multi-task learning. Our final model that combines these two approaches achieves comparable performance with the SportsFormermodel. Thus, we eliminate, the use of three

object detection models, a segmentation model, a vision transformer, and two other transformers from the SportsFormer architecture. Achieving similar performance without the use of basketball specific vision models potentially makes this framework more adaptable to other sports.

3 Related Work

Automated sports commentary generation has been an active area of research, with various approaches proposed to tackle this problem. Some prior work has used computer vision techniques to extract features from sports videos and then used machine learning models to generate commentary. For instance, in [1], player and ball tracking were utilized to generate basketball play-by-play commentary, while in [2], player skeleton detection and ball tracking were employed to generate basketball commentary. While the prior works for basketball video captioning have shown promising results, they typically involve sport-specific modeling techniques that can be complex and difficult to implement. In contrast, our proposed approach avoids basketball specific vision models while still achieving high levels of accuracy. Other papers focus on generating captions for other kinds of sports videos, such as football and volleyball [3].

Transformers have been successfully applied to many tasks in recent years [4]. For example, BERT [5] is a transformer model that has been successfully applied to text input. ViT [6] is a transformer model for images that is the basis for video encoder models such as TimeSformer [7] and VideoMAE [8]. ViT converts an image into a sequence of patches on which a transformer is applied. TimeSformer is the video encoder model used in our paper. Our paper uses transformer models as building blocks for the encoder and decoder.

Transformer models that convert video to text are available in literature. In general, such models are similar to transformer models designed to convert images to text. For example, GIT: A Generative Image-to-text Transformer for Vision and Language [9] is a transformer model that is designed to convert images to text, which can be useful for image captioning. GIT uses an image encoder and a text decoder trained concurrently. End-to-End Transformer Based Model for Image Captioning [10] uses a similar encoder/decoder architecture for image captioning. UniVL [11], develops a pre-trained model for tasks involving video and text. However, generic video to text models such as UniVL do not perform well for sports-specific tasks out of the box.

4 Methods

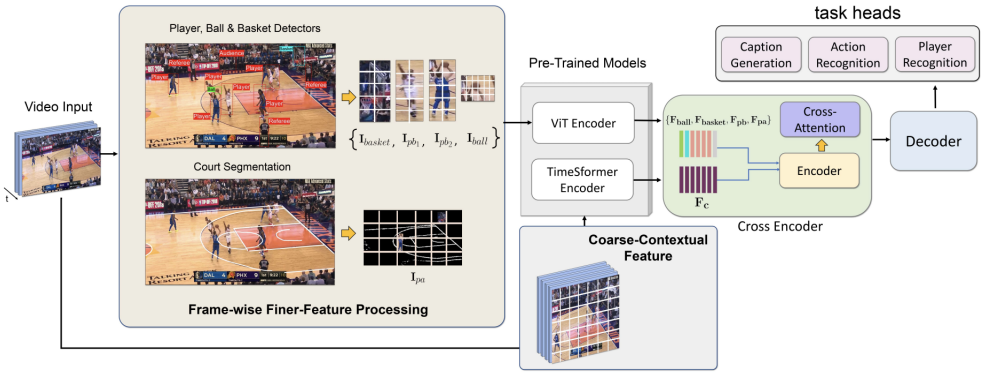

Figure 1: SportsFormer architecture [1].

Initial architecture We base our work on the SportsFormer architecture (Figure 1) proposed by the Sports Video Analysis on Large Scale Data paper [1]. This architecture is largely based on the UniVL architecture [11], and uses the pretrained weights from UniVL. Videos are first fed into TimeSformer [7], a pre-trained video model, to extract spatio-temporal representations. These features are then channeled into a transformer encoder. Additionally, frames from the video are fed into several vision models for ball, player, and basket detection, and court line segmentation, all of which are

channeled into a pre-trained vision transformer model for feature extraction. Then, these features are cross-encoded with the video representation extracted from TimeSformer. Finally, a transformer decoder is used to generate video captions.

When feeding videos into TimeSformer, each frame is decomposed into F non-overlapping patches, each of size $P \times P$, such that the F patches span the entire frame, i.e., $F = HW/P^2$. These patches are flattened into vectors and channeled into several blocks comprised of linear-projection, multi-head self-attention and layer-normalization, in both spatial and temporal axes. The features from the video are defined as

$$F_c = \textit{TimeSformer}(X) \tag{1}$$

where $F_c \in \mathbb{R}^{N*d}$, d is the feature dimension and X is the input clip.

Simplified architecture We identify the following problems with the initial architecture:

1. It requires manual annotation efforts for each of the four basketball vision models (ball, player, basket, and court line models).
2. The models in the pipeline need to be separately trained and evaluated, and the impact of each model on the end-to-end performance requires extensive experimentation.
3. It requires the training of a vision transformer, a transformer to encode the outputs of the vision transformer, and a transformer to cross encode the basketball features with the video features, all of which increase compute requirements and add complexity to the pipeline.

Overall, these problems make this framework difficult to apply to video captioning for other sports. Thus, we propose a simpler pipeline that consists of extracting the video features for each video from our video encoder, and then applying an encoder-decoder model on top of the features. Our encoder is a transformer with 6 layers, where each block has 12 attention heads and the hidden layer size is 768. Our decoder is a transformer with 3 layers, with the same number of attention heads and hidden layer size as the encoder. This modified approach eliminates the use of 3 object detection models, a segmentation model, and two transformer models from the original SportsFormer architecture. Our architecture is shown in Figure 2.

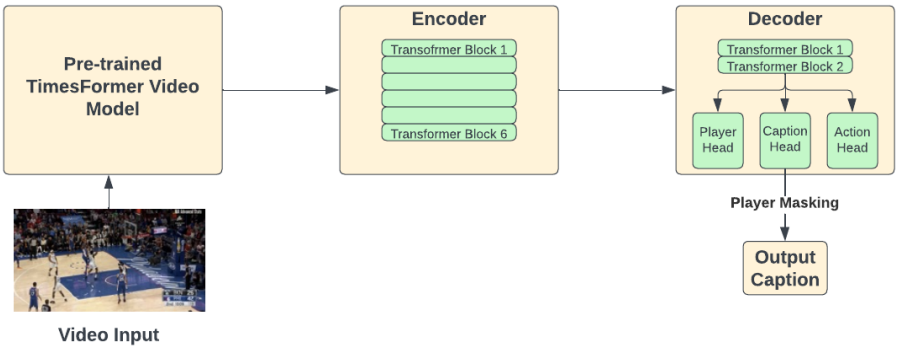

Figure 2: Our final architecture.

Additional data and masking predictions The authors of the SportsFormer paper also experiment with simpler architectures, and the main problem they cite is that it is hard to identify players with Timesformer video features alone (i.e. without the vision models). This often results in captions with player names that are not even in the game.

We use additional data on players to mitigate this. When a sports captioning model is deployed, we will likely have access to the names of the players of the two teams that are playing before the game starts. To add this data to the model, after the final block of the decoder model, we adjust the output logits in the following way:

$$logits = logits + (-10^6\ m), \tag{2}$$

where m is a mask vector. For a given sample (video clip), for all players who are not in the rosters of the two teams in the video, their associated indices are set to 1. All other indices are set to 0. This way, the logits of players on other teams are set to very large negative numbers, and at inference time when the scores are passed through a softmax layer, these players always have a prediction probability of 0. We experiment with applying this masking at evaluation time, and applying it during training and evaluation time.

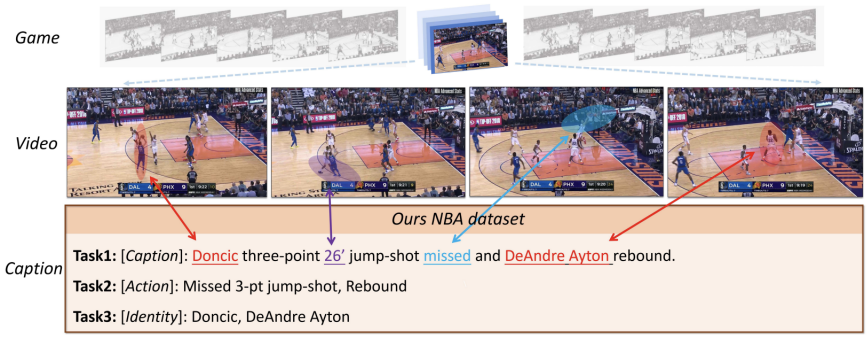

Figure 3: Multitask learning labels

Multi-task learning The dataset provided in the SportsFormer paper also contains lables for the players involved in action and the type of action (three point jumper, dunk, layup, etc.). Figure 3 shows an example of labels for three different tasks generated from a clip. The authors of the SportsFormer paper use their architecture to train two other models on this data: a player identification model, and an action identification model. There is significant overlap between these two tasks and the caption generation task. Multi-task learning can leverage useful information of related tasks to achieve simultaneous performance improvement. Thus, we propose that training a single model with the multi-task learning framework can boost performance for video captioning, particularly for the parts of the caption that relates to players and actions. To implement multi-task learning we do the following:

- Create a custom dataloader such that each batch consists of samples of a single task type (caption generation, player identification, or action identification). This makes the training process fully parallelizable since the whole batch follows the same path through the model.
- For our decoder network, we replace the last block of the transformer with three copies of the decoder block. Thus, each of our tasks share parameters for the encoder and the first two blocks of the decoder, whereas they have their own parameters for the last block of the decoder. The three copies of this last block serve as "task heads" that specialize on the specific task.
- We add functionality to our multitask model such that we can apply the prediction masking described above specifically to the caption generation head.

Baselines Our baseline is the simple architecture we propose in the Simplified Architecture 4 section. The authors of the SportsFormer paper include results about a model similar to this baseline, however the details are not clear and the weights are not publicly available. Therefore, we train this model ourselves.

5 Experiments

Dataset We use the NBA dataset for Sports Video Analysis (NSVA) dataset introduced in the Sports Video Analysis on Large-Scale Data paper [1]. This dataset contains 32,019 video clips, each with associated play-by-play information such as actions and player names. These video clips are from 132 games played by 10 teams in NBA season 2018-2019. Overall, this dataset has higher quality captions than previous datasets in the field, because it uses specific player names and uses various processing methods to make sure the captions don't contain information beyond the scope of the clip.

Contributions to dataset Although the dataset uses player IDs as tokens in captions, it doesn't have information on the whole set of players who could be in court for a given video clip. To apply the method noted in our Approach 4 section and limit the model to predict only the players who could potentially be on court, we use an original scraping script to gather the rosters of the two teams from the NBA website in every clip, and the associated player IDs.

Downstream captioning task evaluation methods Our quantitative evaluation for the video captioning task is based on four widely used NLP evaluation metrics: BLEU, METEOR, CIDEr, and ROUGE-L. The BLEU score [12] measures the n-gram overlap between the generated commentary and a reference commentary, with a higher BLEU score indicating that the generated commentary is more similar to the reference commentary. The METEOR score [13] is similar to the BLEU score, but it takes into account word-level alignments and synonymy between words. CIDEr [14] is a metric designed specifically for evaluating image descriptions. ROUGE-L [15] measures longest common subsequence.

Experiment details For our experiments, we use a BertAdam optimizer and a learning rate $\alpha = 3 * 10^{-5}$ with a weight decay of 0.01. The training time for training each model is 6 hours on a single NVIDIA Tesla T4 GPU. We train each model for 20 epochs using early stopping. At inference, we use beam search with beam size 5.

Results Our results are shown in Table 1. From these results, we can see that each of our incremental changes (player masking during inference, player masking during training, and multi-task learning) improves the model for each of the metrics. We also include the best model from the SportsFormer paper in our table for reference. We can see that our best model is close to the model from this paper, despite our simpler architecture. Specifically, we can see that the CIDEr score is very close between our best model and the SportsFormer best model. Since CIDEr is a metric specifically built for image captioning and uses human consensus, we believe this metric is the most aligned with our task.

Architecture	C	M	B@1	B@2	B@3	B@4	R_L
Modified architecture only	0.977	0.223	0.4840	0.372	0.279	0.211	0.474
Masked eval	1.060	0.227	0.492	0.378	0.284	0.217	0.4781
Masked eval + masked training	1.121	0.229	0.498	0.383	0.290	0.223	0.485
Masked eval + masked training + multi-task learning	**1.138**	**0.233**	**0.506**	**0.393**	**0.298**	**0.230**	**0.491**
SportsFormer paper model [1]	1.139	0.243	0.522	0.410	0.314	0.243	0.508

Table 1: Experimental results of various models run on test data. We did not run the model in the final row, and only include it for reference.

6 Analysis

Qualitative comparison of captions Table 2 shows an example caption generated by three different models, as compared to the ground-truth reference. For the first model (modified architecture only), we see that the model correctly identifies that there is a missed shot and a rebound. However, the type and distance of the missed shot is incorrect, and the player that missed the shot is incorrect (in fact, it is a player not in the game), and the player for the defensive rebound is incorrect. Additionally, the caption does not include the offensive rebound. Moving on to the masked eval + masked training model, we expect an improvement in player names. This indeed happens, where the player who missed the shot is corrected. However, the type of shot is still incorrect, the defensive rebounder is incorrect, and the caption still doesn't include the offensive rebound action. Finally, moving on the masked eval + masked training + multi-task learning model, we expect a possible improvement in actions in the caption. This indeed happens in two ways. The type and distance of the missed shot is corrected (1 ' tip layup instead of 4 ' layup). Also, the offensive rebound action and player is correctly generated. The generated caption is still not correct, because the defensive rebounding player is

Model	Caption
Modified architecture only	Miss LaMarcus Aldridge 4' layup Rudy Gobert defensive rebound
Masked eval + masked training	Miss Rudy Gobert 4' layup Rudy Gobert defensive rebound
Masked eval + masked training + multi-task learning	Rudy Gobert offensive rebound Rudy Gobert 1' tip layup shot Rudy Gobert defensive rebound
Reference	Rudy Gobert offensive rebound Miss Rudy Gobert 1' tip layup shot Steph Curry defensive rebound

Table 2: Example comparison of a predicted caption across different models.

incorrect. Overall, this example demonstrates that the intuition behind our progressive improvements to the model leads to expected improvements with some limitations. We address these limitations in the next section.

Performance breakdown by action type For our best model (i.e. the model with player masking and multi-task learning), we investigate which actions perform the best when comparing our predicted captions to the ground truth captions. In order to do this, given an action a and a caption c, we define $\mathbb{I}(c, a)$ as true when c contains a and false otherwise. Using this function, we calculate the precision and recall of $\mathbb{I}$ for our model compared to the ground truth $\mathbb{I}$. Our results are shown in Figure 3. We can see that the precision and recall for "shot" are relatively high, meaning that our model is successful at including "shot" in the predicted caption. However, the precision and recall for "layup shot" are much lower. This indicates that the model has a harder time identifying layup shots compared to normal shots. This intuitively makes sense since layup shots are a subset of shots and are harder to identify. This is in general true for actions, and the performance degrades the more specific an action is. For example, performance on offensive rebound is lower than the performance on rebound in general. Furthermore, we see that "foul" and "rebound" have high precision and recall as well, while "turnover," which is a rarer action has lower precision and recall.

Action	Precision	Recall
shot	0.774	0.751
layup shot	0.438	0.496
foul	0.768	0.827
rebound	0.753	0.824
turnover	0.618	0.508

Table 3: Precision and recall for various actions between our best model and the ground truth captions.

Common failures One common failure we see from our model is its tendency to predict the same name multiple times in the caption, even in places it does not make sense. For example, for one video, our best model predicts "jump ball LaMarcus Aldridge vs. LaMarcus Aldridge" for one video. This is impossible, since in basketball one cannot have a jump ball with themselves. Another common failure for our model is predicting the distance of shots. For example, our model will predict a 14 foot jump shot when the true caption is a 21 foot jump shot. We believe that getting the distances correct for shots is difficult for our model since the video model downsamples the videos, and therefore the model loses spatial information.

7 Conclusion

In this paper, we have demonstrated that with careful use of data, modifications to the decoder architecture of the previous work and a multi-task learning framework, we can have a simple and high performing model for basketball video captioning. Our best model is composed of a pre-trained video transformer, a pre-trained encoder to process the video encodings, and a decoder model. This approach eliminates the use of three object detection models, a segmentation model, a vision transformer, and two other transformers from the architecture. Our method removes the need for training basketball specific vision models, which potentially makes this framework more adaptable to other sports.

Future Work The main bottleneck in our model's performance is how the video features are generated. The TimesFormer video model is pre-trained on a large corpus of videos, but it is not fine-tuned on basketball specific videos. Finetuning a video model such as on basketball videos with unspervised learning could significantly increase the quality of the video features and increase performance. Additionally, using TimesFormer requires using a smaller resolution, and this causes the model to lose spatial information. Using or developing a video model that can process larger resolution frames would increase performance.

A final avenue for future work, and perhaps the most exciting, is more creative caption generation. The NSVA dataset that we used has mechanistic play-by-play captions that would not be sufficient to replace human commentary. The main challenge for this task is that human commentary includes information that is not available from the video clip alone, and often is not time-aligned with the video. Creative caption generation would require the creation of a new dataset and possibly a more complex modeling approach.

8 Other Information

Here is the code for this project: https://github.com/lucaspauker/NSVA.

References

[1] Dekun Wu, He Zhao, Xingce Bao, and Richard P. Wildes. Sports video analysis on large-scale data, 2022.

[2] Huanyu Yu, Shuo Cheng, Bingbing Ni, Minsi Wang, Jian Zhang, and Xiaokang Yang. Fine-grained video captioning for sports narrative. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6006–6015, 2018.

[3] Mengshi Qi, Yunhong Wang, Annan Li, and Jiebo Luo. Sports video captioning via attentive motion representation and group relationship modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2617–2633, 2020.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *CoRR*, abs/2102.05095, 2021.

[8] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022.

[9] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language, 2022.

[10] Yiyu Wang, Jungang Xu, and Yingfei Sun. End-to-end transformer based model for image captioning, 2022.

[11] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *CoRR*, abs/2002.06353, 2020.

[12] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[13] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. pages 228–231, 07 2007.

[14] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014.

[15] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.