Automated Basketball Video Captioning

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

Abstract

This paper presents a simple model for generating play-by-play captioning of basketball videos using foundation models. Previous work has been done on this problem typically it involves complex basketball-specific modeling approaches. In contrast, our proposed model aims to be simpler and sport-agnostic. Our model consists of two parts: a video encoder and a text decoder, and we experiment with methods of improving each. We find that TimeSformer [1] and VideoMAE [2] video encoders perform similarly, and our model's text decoder is improved by adding masking player tokens and multi-task learning. Our model is simpler than other models in the literature, has comparable performance, and can quickly generate basketball captions.

1 Introduction

In this paper, we investigate automated captioning of basketball videos using foundation models. 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.

Although some previous work has been done on this problem [3, 4], 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.

2 Related Work

There are many video encoder models available in the literature. For example, TimeSformer [1] is a transformer-based video encoder that is based on the Vision Transformer (ViT) [5] architecture. In recent years, there has been a growing interest in using self-supervised learning techniques to process videos. Using self-supervision allows models to be pretrained on large corpuses of unlabeled video data. VideoMAE [2] is a transformer model for videos that uses joint time-space attention and can learn in a self-supervised fashion. The authors demonstrate that the model can be pretrained using self-supervised learning by learning to reconstruct masked out pixels. This is similar to the approach taken for images in VitMAE [6]. In this paper, we experiment with TimeSformer and VideoMAE for our video encoder.

In addition, there are transformer models specifically designed to convert images to text. For example, GIT: A Generative Image-to-text Transformer for Vision and Language [7] 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 [8] uses a similar encoder/decoder architecture for image captioning. UniVL [9], develops a pre-trained model for tasks involving video and text. In this paper, we use an encoder/decoder

architecture for our video captioning problem, however we train the encoder separately from the decoder.

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 [3], player and ball tracking were utilized to generate basketball play-by-play commentary, while in [4], 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 leverages foundation models to simplify the modeling process and make it more efficient while still achieving high levels of accuracy.

3 Methods

Initial architecture We base our work on the Sportsformer architecture (3) proposed by the Sports Video Analysis on Large Scale Data paper [3]. This architecture is largely based on the UniVL architecture [9], and uses the pretrained weights from UniVL. Videos are first fed into TimeSformer [1], 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.

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. Overall, our model for automated captioning for basketball videos can be divided into a video encoder network and a language decoder network. We will describe each of these separately.

3.1 Video Encoder

Self-supervised finetuning We use self-supervised learning to finetune the baseline VideoMAE model trained on the Kinetics-400 dataset [10]. We use a mask reconstruction task to finetune the model on our data. In the case of image masked autoencoders, the self-supervised learning process involves masking a subset of tokens from an input image and feeding the remaining tokens into a transformer encoder. Then, a shallow decoder is used to reconstruct the image. In the case of video masked autoencoders, the design is customized to address the characteristics of video data, including temporal redundancy and temporal correlation. VideoMAE uses a cube embedding layer to decrease the spatial and temporal dimension of the input, and it uses a temporal tube masking mechanism to encourage learning of spatiotemporal structure representations. Tube masking is a masking strategy where the same masking map is applied to all frames in a video clip. We run experiments with finetuning with different mask ratios.

3.2 Text Decoder

Masking player tokens The Sportsformer paper experimented with simpler architectures for identifying players in videos, but found that they often produced captions with incorrect player names, particularly without the aid of task-specific vision models. To overcome this limitation, we propose using additional data on the players, such as their names and teams.

To incorporate this data, we apply a mask to the output logits at the final block of the decoder. The mask is generated by setting the indices of players not in the rosters of the two teams playing in the video to 1, and all other indices to 0. This sets the logits of players on other teams to very large negative numbers, resulting in a prediction probability of 0 at inference time when passed through a softmax layer. We experiment with applying this mask during both training and evaluation or only at evaluation time.

Multi-task learning The Sportsformer paper trained player and action identification models alongside their caption generation model. As these tasks share similarities, multi-task learning can improve performance by leveraging related information. Therefore, we suggest multi-task learning to enhance video captioning, with a particular focus on improving the recognition of players and actions.

To implement multi-task learning, we modify the decoder network with "task heads" that specialize in each task. Specifically, the encoder and the first two blocks of the decoder share parameters, whereas the last decoder block has its own parameters for each task. This allows the model to utilize the shared knowledge and learn task-specific features simultaneously.

4 Experiments

Dataset We use the NBA dataset for Sports Video Analysis (NSVA) dataset introduced in the Sports Video Analysis on Large-Scale Data paper [3]. 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.

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 [11] 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 [12] is similar to the BLEU score, but it takes into account word-level alignments and synonymy between words. CIDEr [13] is a metric designed specifically for evaluating image descriptions. ROUGE-L [14] measures longest common subsequence.

4.1 Finetuning Video Model

We use self-supervised learning with unlabeled basketball videos to finetune the VideoMAE [2] model. For our experiments, we use an Adam optimizer with a learning rate of $\alpha = 2 * 10^{-4}$. It took 9 hours to train each model for 3 epochs on a single NVIDIA Tesla T4 GPU.

Masking ratio The main parameter we experimented with altering was m, the ratio of pixels that are masked. Note that since we are using a tube masking approach, the same pixels are masked for all timesteps in the video. We experiment with $m = 0.50$, $m = 0.75$, and $m = 0.90$. The training losses for each of these masking ratios can be seen in Figure 1. We find that the lower masking ratio learns faster, with $m = 0.50$ yielding the lowest training loss after 3 epochs. The authors of the VideoMAE paper find that higher masking ratios such as $m = 0.90$ and $m = 0.95$ work the best. Our results show that lower values of m are better in terms of training loss, however the result may be different if we trained for more epochs.

Visualization In order to compare the features learned by the different vision models for all the basketball videos, we use PCA to reduce the dimension of the features to two. Figure 2 shows the

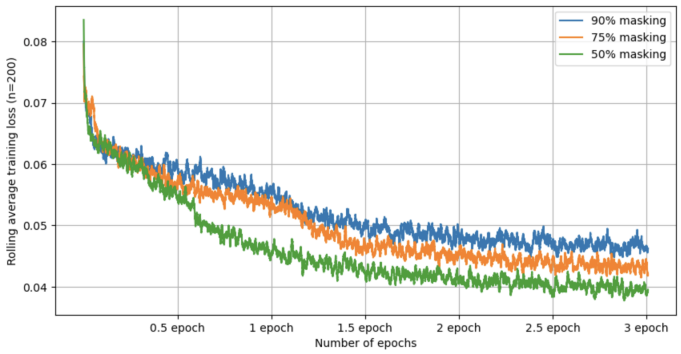

Figure 1: Training loss curves for VideoMAE model self-supervised pre-training on basketball video data with different masking ratios.

results of the top two PCA features graphed against each other for the TimeSformer, VideoMAE, and finetuned VideoMAE models. Each graph shows where videos from different games lie in this reduced-dimensional space. The not-finetuned VideoMAE features are randomly distributed and there is no clear distinction between games. For the finetuned VideoMAE and TimeSformer features, however, there are clear groupings of points. This suggests that the feature representations for TimeSformer and finetuned VideoMAE models are better than the not-finetuned VideoMAE features. Comparing TimeSformer and the finetuned VideoMAE features, we can see that the finetuned VideoMAE distinctions between groups of games is a little more sharp.

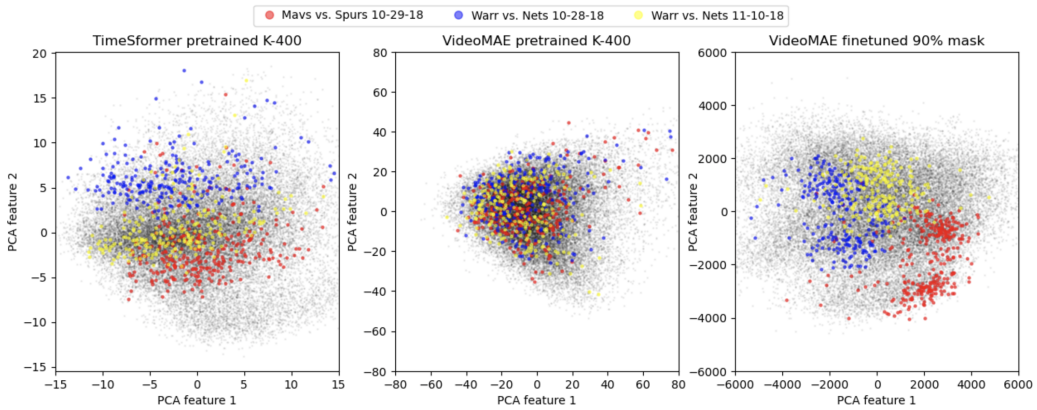

Figure 2: PCA feature comparisons on NSVA dataset for TimeSformer, VideoMAE, and finetuned VideoMAE models. Each point represents a separate video and points are colored for three games.

Downstream tests In order to use the video models in downstream applications, we extract features from each video with our model then feed them into our text decoder. In order to extract features, we first downsample the video. Given video with f frames, we use a sampling rate $s = 8$, then downsample the video to have $f' = f/s$ frames. Since the pretrained vision model takes an input of n frames (i.e. for VideoMAE, $n = 16$), we feed in n frames from the downsampled video into the model at a time and do average pooling to get a vector of length k, where k is the hidden size of the model. Thus, our features for a given video are a vector in $\mathbb{R}^{\frac{f'}{n} \times k}$.

From Table 1, we can see that the TimeSformer model pretrained on the Kinetics-400 dataset outperforms and the VideoMAE model pretrained on the Kinetics-400 dataset perform similarly. VideoMAE has a slight edge, however, when masking player tokens and multi-task learning are applied. We see that our self-supervised finetuned VideoMAE models perform worse than the not-finetuned VideoMAE model. This may be because the features learned in finetuning are not the same features that are useful for generating captions, such as the player names and actions taken. We find

that the finetuned VideoMAE model with 50% masking performs better on the downstream task than the model with 90% masking.

4.2 Text Modeling

For our text 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.

Masking player tokens and multi-task learning In order to evaluate the effectiveness of masking player tokens and multi-task learning, downstream captioning performance for the TimeSformer and VideoMAE features with and without masking player tokens and multi-task learning. Our results are shown in Table 1. From these results, we can see that for both the TimeSformer and VideoMAE features, the addition of masking player tokens and multi-task learning significantly improves all of our metrics. Notably, the CIDEr score improves significantly by adding masking player tokens and multi-task learning. 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.

Video Features	Architecture	C	M	B@1	B@2	B@3	B@4	R_L
TimeSformer pretrained on Kinetics-400	MA	0.900	0.212	0.459	0.352	0.262	0.198	0.452
	MA+MM	1.035	0.216	0.476	0.365	0.274	0.211	0.461
VideoMAE pretrained on Kinetics-400	MA	0.885	0.210	0.454	0.347	0.257	0.192	0.449
	MA+MM	1.091	0.224	0.494	0.380	0.285	0.219	0.477
VideoMAE finetuned with 90% masking	MA+MM	0.694	0.179	0.390	0.287	0.211	0.160	0.3626
VideoMAE finetuned with 50% masking	MA+MM	0.736	0.187	0.411	0.300	0.218	0.163	0.390
TimeSformer	Reference [3]	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. MA=modified architecture, MM=masking tokens and multi-task learning.

5 Conclusion/Future Work

In this paper, we build a model for basketball caption generation by leveraging foundation models. First, we simplify the Sportsformer architecture to reduce the number of models in the pipeline. Then, we investigate changing the video encoder as well as the text decoder networks to improve performance. We find that the TimeSformer and VideoMAE video encoders perform similarly. We also finetune VideoMAE in a self-supervised fashion on our basketball video data and find that although the feature representations are better, this does not improv e downstream performance. For the text decoder, we add masking player tokens and multi-task learning to improve performance on our quantitative metrics. Overall, our proposed model is simpler than other models in the literature, has comparable performance, and can quickly generate basketball captions.

For future work, we would like to understand more deeply why the finetuned VideoMAE models do not perform well on the downstream task. We would also like to explore using a larger dataset of unlabeled basketball videos to improve the finetuned models. Lastly, it would be interesting to extend our model to other sports domains and explore its applicability in other areas, such as surveillance and video summarization.

6 Other Information

I am using this project topic for CS224N as well. I worked with my CS224N partner on the language modeling (NLP) parts and did the vision portion of the paper myself. Here is the code for this project: https://github.com/lucaspauker/NSVA.

References

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

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

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

[4] 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.

[5] 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.

[6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[7] 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.

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

[9] 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.

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[11] 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.

[12] 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.

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

[14] 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.

7 Appendix

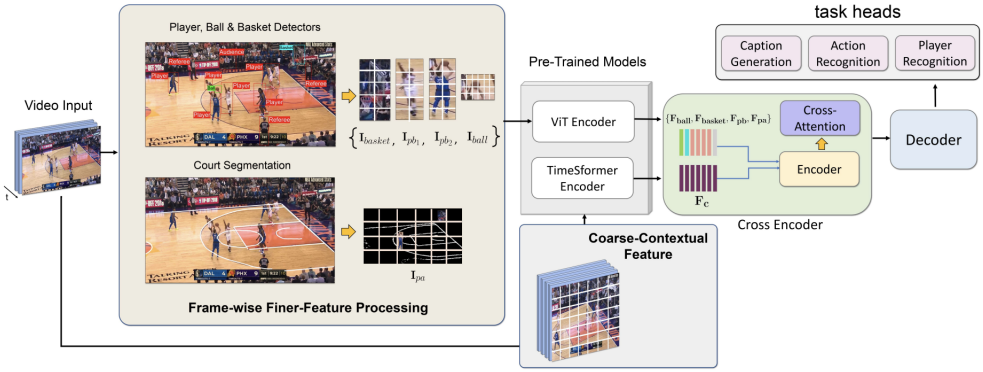

Figure 3: Sportsformer architecture [3].