CAMBATE: Video Captioning with Mamba-Transformer

Ng Jing Xu
Peking University
2100094806@stu.pku.edu.cn

Abstract

Video captioning aims to generate associated captions according to the video's temporal and spatial features. CAMBATE is an end-to-end video captioning model with Mamba and Transformer. CAMBATE aimed to create a model that generates accurate and coherent captions for video content. Key components of the project included data processing, model development, training and visualizing. Experimental results show that the model performs reasonably well.

1. Introduction

Video captioning is an essential tool in today's digital age, enhancing accessibility and usability of video content across various domains such as education, entertainment, and news. The ability to generate accurate subtitles not only improves the viewer experience but also ensures content is accessible to a wider audience, including those with hearing impairments. CAMBATE focuses on developing a deep learning-based system with Mamba and Transformer, that can generate captions for video content, leveraging state-of-the-art CV and NLP techniques.

2. Related work

2.1. Dense Video Captioning

Dense video captioning is a task in computer vision that involves generating descriptive captions for multiple events or segments within a video. Unlike traditional video captioning, which typically generates a single caption for an entire video, dense video captioning aims to provide a detailed and comprehensive understanding of the video content by identifying and describing all significant events.

2.2. Convolution 3D

The Convolutional 3D (C3D) network is a pioneering deep learning model for video analysis, capturing spatiotemporal features from videos by employing 3D convolutions. It extends the concept of 2D convolutions to the temporal dimension. The generic nature of C3D features makes them suitable for various video classification tasks, including object recognition, scene classification, and action recognition. This extension allows for the extraction of rich spatio-temporal features from video clips, capturing motion information critical for accurate caption generation. Previous studies have shown that C3D effectively models temporal dynamics, making it a suitable choice for this project.

2.3. State Space Model (Mamba)

Mamba is a sequential model which integrate selective state space models (SSMs) into a simplified end-to-end neural network architecture without attention or MLP blocks. It has shown significant effectiveness of state space transformation in capturing the dynamics and dependencies of language sequences. Besides, it was specifically designed to model long-range dependencies, boasting the advantage of linear complexity. Recently, the great potential of Mamba has also been led into vision tasks, demonstrating better performances and higher GPU efficiency than Transformer on visual downstream tasks like object detection and semantic segmentation.

2.4. Transformer

The Transformer architecture has revolutionized NLP tasks with its self-attention mechanism and parallel processing capabilities. In the context of video captioning, Transformers are used to decode the encoded video features into coherent textual descriptions. Their ability to model long-range dependencies is particularly beneficial for generating captions that accurately reflect the video content.

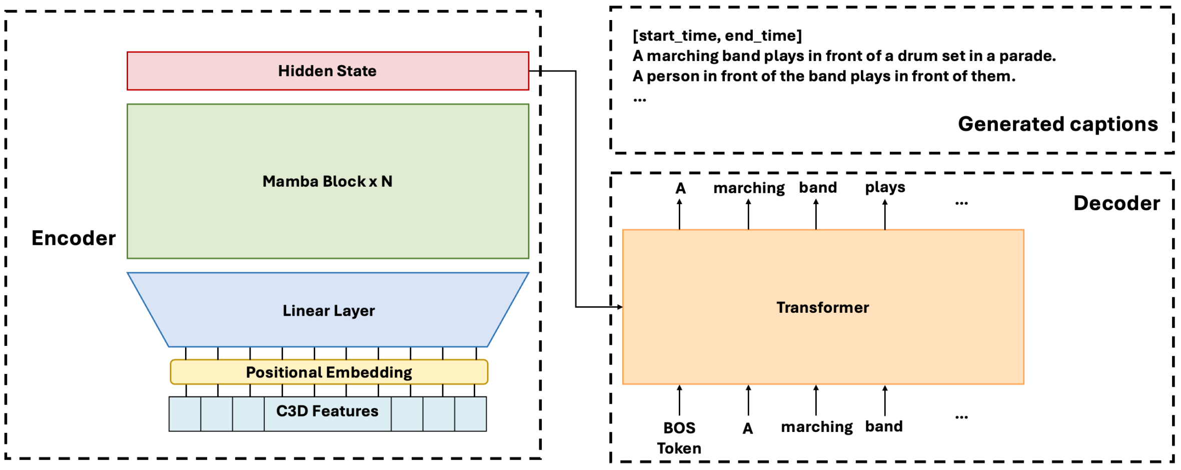

Figure 1: Overview of CAMBATE video captioning model

3. Method

3.1. Dataset

The feature dataset is a matrix of C3D features (feature frame, hidden dimensions), with 500 hidden dimensions for every 8 frames. Thus, the size of a feature frame can be counted by:

$$Number\ of\ feature\ frame = \frac{\text{Duration of video(second)} \times \text{Frames per second}}{8}$$

For the ease of training model, split the features into batches with 32 feature frames. Besides, in order to establish the model which can see the context before an action happened, each batch is lengthen at front with 32 more feature frames of the previous batch, and pad a blank context for the first batch. Batches now contain 64 feature frames, while total number of batches remain the same. Thus, there is an overlap of 32 feature frames for every batch. Figure2 illustrates the process of overlapping feature frames.

Captions dataset is a dictionary for every video, index with video id, list of captions and the timestamps for each caption in second. The captions are assigned to the associated batches if the timestamp of the captions overlap with the timestamp of the batches. If there are two captions overlap the same batch at the same time, the captions are connected sequentially. Figure3 illustrates the process of concatenating captions.

3.2. Mamba Encoder

The encoder, comprising Mamba layers, processes the features extracted by the C3D network. It generates a high-dimensional representation of the video content, capturing both spatial and temporal information. This encoding step is crucial for generating accurate captions. These layers have been shown to enhance the model's ability to learn and represent intricate video content, which is crucial for generating meaningful captions.

3.3. Transformer Decoder

The Transformer decoder is tasked with generating textual descriptions from the encoded video features. Utilizing self-attention mechanisms, the decoder focuses on relevant parts of the video representation and generates words sequentially to form coherent sentences. The generation process involves predicting the next word in the sequence until a complete sentence is formed. Beam search is employed to optimize word selection, ensuring the generated captions are both accurate and fluent.

3.4. Beam Search Generation

Beam search is a search algorithm used in the decoding phase to generate sequences of words. It explores multiple hypotheses simultaneously, balancing between breadth and depth of search to optimize the generation of fluent captions. It ensures that the most likely sequence of words is selected, leading to more accurate and coherent captions.

3.5. Tokenizer

The GPT-2 tokenizer is used to tokenize the input and output text. It converts words into tokens that can be processed by the model and reconstructs tokens into human-readable text after generation. The use of GPT-2 tokenizer ensures consistency and compatibility with existing models.

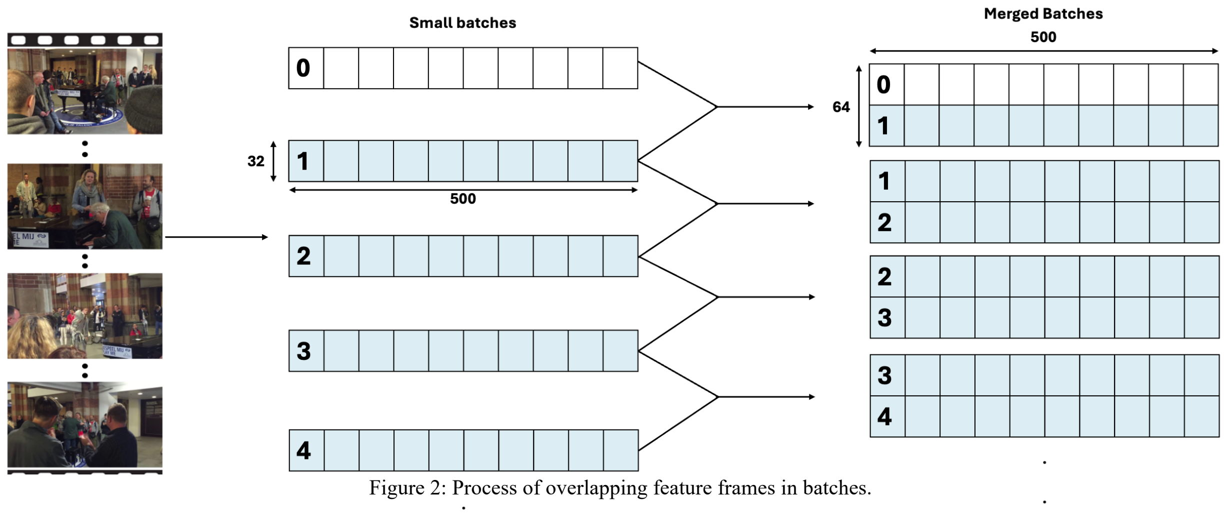

Figure 2: Process of overlapping feature frames in batches.

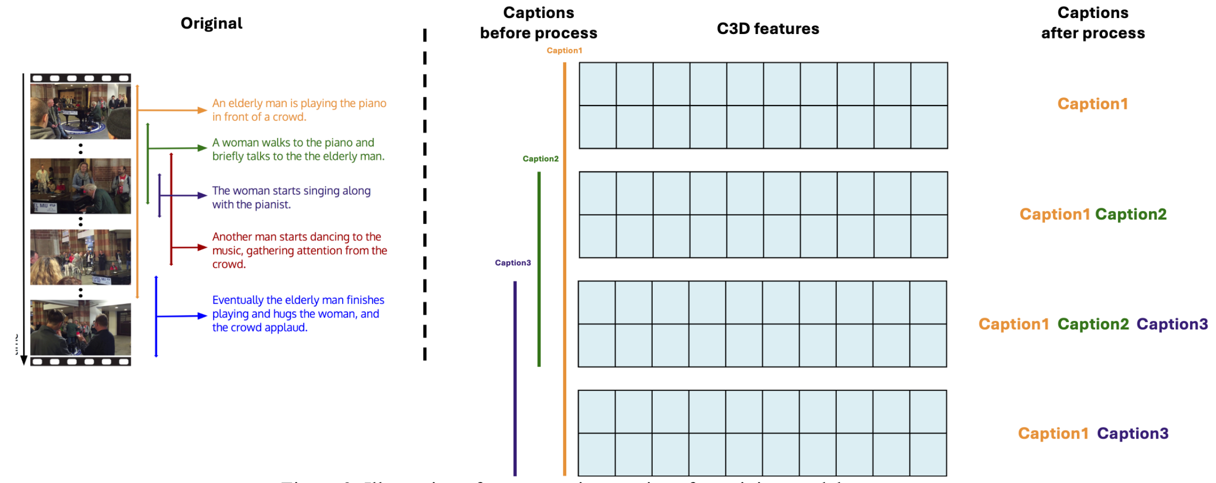
Original
An elderly man is playing the piano in front of a crowd.
A woman walks to the piano and briefly talks to the the elderly man.
The woman starts singing along with the pianist.
Another man starts dancing to the music, gathering attention from the crowd.
Eventually the elderly man finishes playing and hugs the woman, and the crowd applaud.
Captions before process
C3D features
Captions after process
Caption1
Caption2
Caption3
Caption1
Caption1 Caption2
Caption1 Caption2 Caption3
Caption1 Caption3

Figure 3: Illustration of concatenating captions for training model.

4. Experiment and Results

The experimental setup involved training the model on the ActivityNet Captions dataset, which contains a diverse range of video content and corresponding captions. It contains 20k videos amounting to 849 video hours with 100k total descriptions, each with its unique start and end time.

The training process was conducted on a GPU to accelerate computation. With 4 A100 Nvidia GPUs (80GB), the total training time is 70 minutes, with the total consumption of 140GB memory averagely.

Mamba encoder was stacked 12 layers, with a total of 130M trainable parameters. Transformer decoder was stacked 4 layers, with 115M trainable parameters. For every video feature, it split into a shape of [Batch, 64, 500] and input to the Mamba encoder. Mamba encoder generates [Batch, 64, 768] hidden states. The hidden states are decoded by Transformer decoder, one caption which contains several sentences are generated for each batch.

During the generation process with the video id and encoder decoder model given, it separate the video feature into multiple batches with 64 feature frames and 500 hidden dimensions. There are no overlapping while generating captions for the video. In order to visualize the generated caption easily, each generated caption is limit to 3 sentences. Which means that 3 sentences are generated for a total of 64*8 = 512 frames (nearly 21 seconds for videos with 25fps). These 3 sentences split the timestamp evenly. For example, if the timestamp for caption1, caption2, caption3 are [1,21], they will split the timestamp into caption1 shows on [1,7], caption2 shows on [8,14], caption3 shows on [15,21].

The generation results are shown in Figure4.

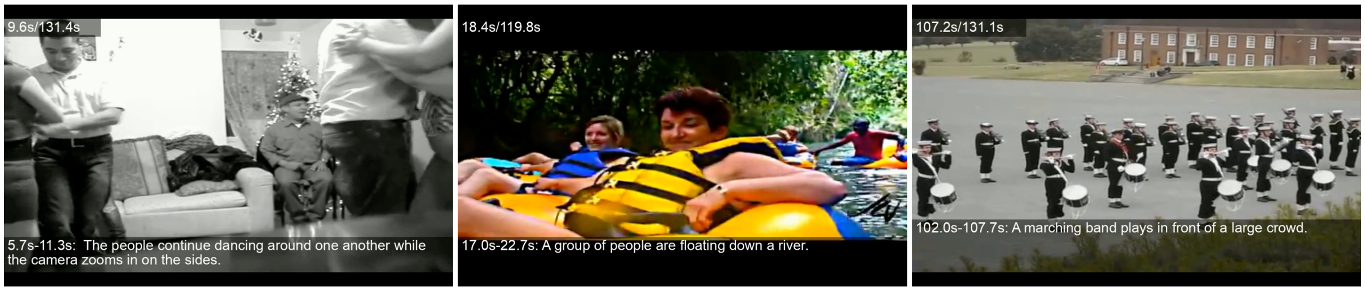
9.6s/131.4s
5.7s-11.3s: The people continue dancing around one another while the camera zooms in on the sides.
18.4s/119.8s
17.0s-22.7s: A group of people are floating down a river.
107.2s/131.1s
102.0s-107.7s: A marching band plays in front of a large crowd.

Figure 4: Some results of generated captions with videos.

5. Tasks done in this project

5.1. Data

- Downloads C3D features and captions
- Split the features into shape of [Batch, 64, 500], with 32 feature frames overlapping
- Concatenate captions according to their timestamp with the features

5.2. Encoder

- Modified from mamba_ssm/models/mixer_seq_simple.py
- Learnable position embedding layer
- Linear layer which project the position embedded input features to higher dimension
- Stack 12 layers of Mamba blocks
- Apply layernorm

5.3. Decoder

- Accept output from encoder
- Embed the vocab size into hidden state size
- Add position embedding
- Stack 4 layers of Transformer block
- FC layer project hidden states to vocab

5.4. Tokenizer

- Uses gpt2 tokenizer

5.5. Train

- Create dataloaders
- Apply Cross Entropy Loss, Adam optimizer, StepLR scheduler
- Run 10 epochs
- For every video feature, tokenize the ground truth caption
- Forward pass with encoder
- Create shifted target for the decoder
- Forward pass with decoder
- Calculate loss, back propagation
- Save encoder and decoder model for every epoch
- Save the loss value of every epoch

5.6. Test (generation)

- Load checkpoint files for MambaEncoder and TransformerDecoder
- Create dataloaders
- A video link is printed, user can download the video for visualization
- Use beam search algorithm to generate captions with max length of 100 tokens
- Captions are written into file (output.txt and visualization/results.json)

5.7. Visualization

- Modified from PDVC/visualization/visualization.py
- Load every frames from the video
- Paint the caption onto the video
- Output a video with caption applied
- Done with multithread (Using 8 threads to paint the caption has the fastest performance)

6. Possible future works

6.1. Adding C3D Net

This work can be extended by adding a conv3d net and pre-trained weights. It is the main problem for adding more training datasets as the videos are unable to convert into C3D video features, which will consume lower resources when training. This problem also act as an obstacles to let the user generate captions on own video.

6.2. Varies parameters

Varies the number of layers, size of hidden dimensions and evaluate their results. Current configurations of model need high computation cost. The same result or even better result might be achieved by lowering the number of layers and size of hidden dimensions. It might has a sky-rocketed performance if increase these parameters. Evaluations for different configurations was not done in this project and the limit of this model remains unknown.

7. Conclusion

This paper presents CAMBATE, an end-to-end video captioning model with Mamba blocks and Transformer blocks. This model demonstrates the potential of deep learning and NLP techniques in generating accurate and coherent captions for video content. The generation results are visualizable.

References

[1] Gu, A. and Dao, T., "Mamba: Linear-Time Sequence Modeling with Selective State Spaces", <i>arXiv e-prints</i>, 2023. doi:10.48550/arXiv.2312.00752.

[2] Vaswani, A., "Attention Is All You Need", <i>arXiv e-prints</i>, 2017. doi:10.48550/arXiv.1706.03762.

[3] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M., "Learning Spatiotemporal Features with 3D Convolutional Networks", <i>arXiv e-prints</i>, 2014. doi:10.48550/arXiv.1412.0767.

[4] Wang, T., Zhang, R., Lu, Z., Zheng, F., Cheng, R., and Luo, P., "End-to-End Dense Video Captioning with Parallel Decoding", <i>arXiv e-prints</i>, 2021. doi:10.48550/arXiv.2108.07781.

[5] Dense-captioning events: https://cs.stanford.edu/people/ranjaykrishna/densevid/

[6] C3D video features: https://rochester.app.box.com/s/8znalh6y5e82oml2lr7to8s6ntab6mav/folder/137471953557

[7] Blog for Mamba: https://www.maartengrootendorst.com/blog/mamba/