CS 231N Short Form Video Captioning with C3D and CLIP

Chenyang Dai (qddaichy@stanford.edu)

Objectives

The goal of this project is building video captioning system for the short form videos focusing on MSVD and MSR-VTT dataset. We mainly have three contributsions:

1. Implement and train two encoder-decoder models using C3D and CLIP encoding while both models use GPT-2 for deocoding.
2. Conduct various experiments regarding training strategies and find that pretraining with image captioning and fine tuning specific layers can greatly boost performance..
3. Conclude that adopting beam search, data augmentation and use more input frames can further improve the performance.

C3D Encoding

The C3D encoding model consists of 8 3D convolutional layers, 5 pooling layers and 3 fully connected layers. The final fully connected layers have a size of 15360 dimensions, which is then tokenized into 20 GPT-2 context tokens. GPT-2 uses context tokens to conditionally generate captioning in natural language.

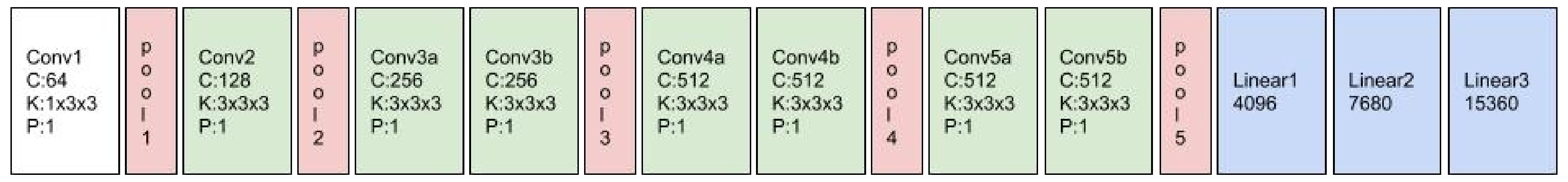

Figure 1: C3D based encoder structure

CLIP Encoding

For CLIP encoding model, we have custom logic to parallel process video frames. There is no fusion logic for different frames' CLIP feature vectors. We have a custom MLP layer that directly maps each frame feature to 4 GPT-2 tokens (4 x 5 frames = 20 tokens in total). The reason behind is that we want to defer the fusion logic to GPT-2 who has a lot more attention blocks and layers that can gradually fuse the in formation together.

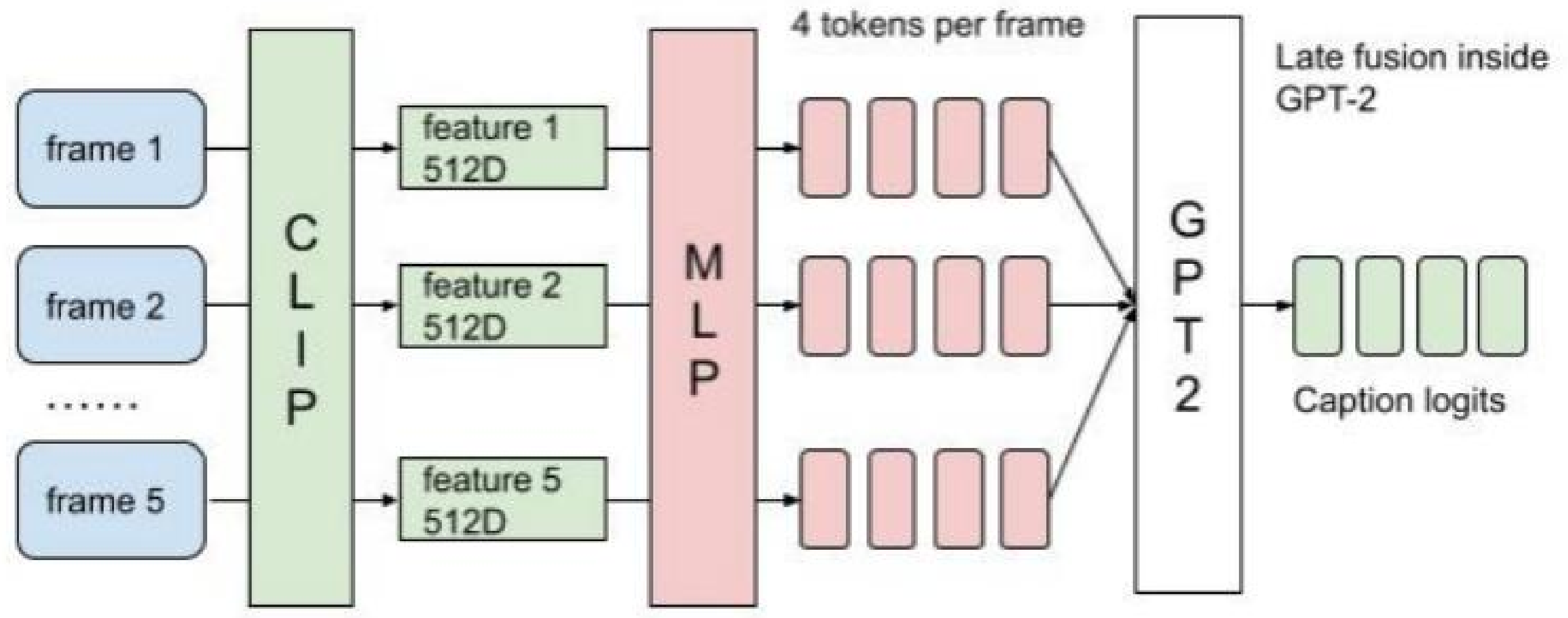

Figure 2: CLIP based encoding model

GPT-2 Decoding and Loss Function

We concatenate the 20 context tokens with the tokenized ground truth captioning text to from one training example. Then we pad all examples to the same length. For each example, we build the corresponding attention mask allowing the model to pay attention to context tokens while masking out all future tokens or pad tokens

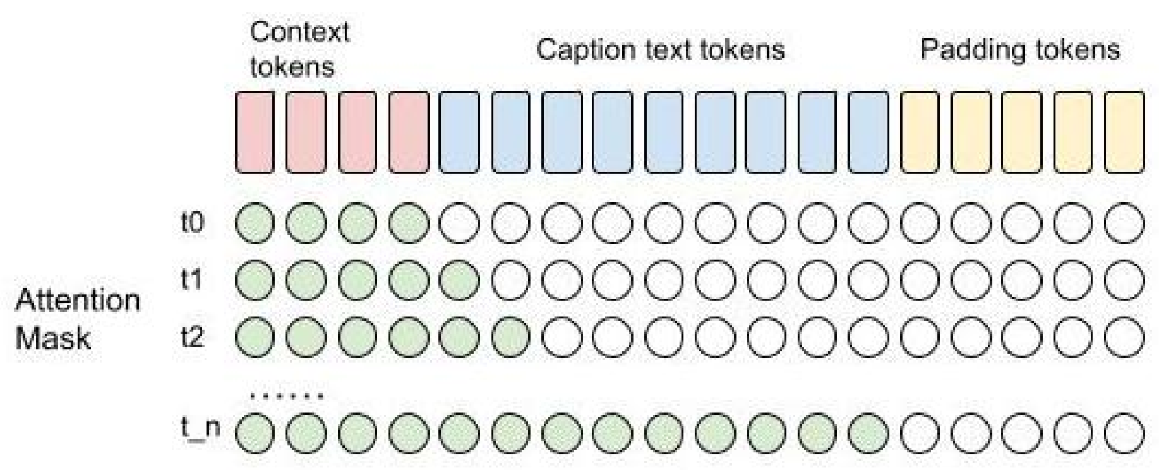

Figure 3: GPT-2 custom attention mask

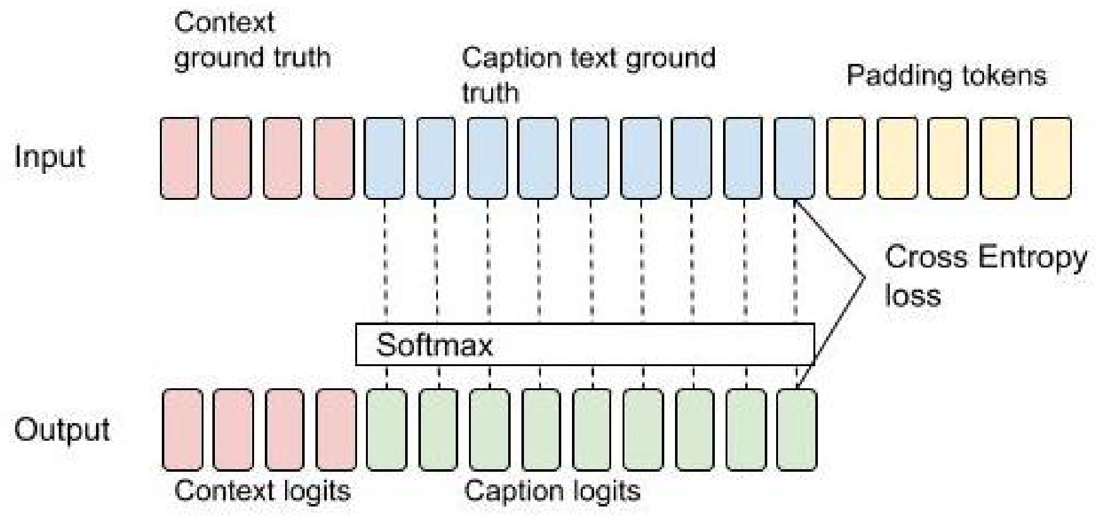

Figure 4: Model loss calculation

We manually calculate the loss disregarding the first 20 positions in the output logits. Our final loss is calculated by only comparing the captioning text part.

Experiment Results

Best model prformance comparsion

Model	MSVD (BLEU-4)	MSR-VTT (BLEU-4)
CNN-LSTM (2015, baseline)	33.3	
PickNet (2018)	46.1	38.9
GRU-EVE (2019)	47.9	38.3
STG-KD (2020) [27]	52.2	40.5
SWIN BERT (state of the art)	58.2	41.9
CLIP+GPT2 (Our Model)	**46.39**	**36.45**

Table 1: Performance comparions with previous methods

	Validation Set	Test Set
No pretrain C3D + GPT-2	34.08	30.67
No pretrain CLIP + GPT-2	**37.14**	**33.54**

Table 2: Comparison between C3D and CLIP encoding

CLIP encoding outperforms C3D encoding in our experiment . We believe that it is because of the power of CLIP's large scale pre-training.

	MSVD Valida-tion B-4	MSVD Test B-4	MSR-VTT Test B-4
No pretrain CLIP + GPT-2	40.97	37.70	28.60
Pretrain CLIP + GPT-2	**46.37**	**42.32**	**32.27**

Table 3: Question categories breakdown

Pretraining boosts performance by a very large margin. The 80k COCO training data can better generalize the model.

Fine tune Strategy	MSVD Validation	MSVD Test	Training Time per epoch
Only MLP	37.14	33.54	~ 6 mins
MLP + GPT-2	40.97	**37.70**	~ 7 mins
CLIP + MLP + GPT-2	**42.65**	37.38	~ 38 mins
CLIP + MLP	39.32	34.10	~ 38 mins

Table 4: Results for fine tuning different layers

Fine tuning CLIP takes longer. Fine tuning GPT-2 can greatly improve performance where as fine tuning CLIP does not benefit performance much.

Method	MSVD Val B-4	MSVD Test B-4
Greedy Search	46.37	42.32
Beam search (width =3)	48.11	44.44
Beam search (width =4)	49.96	45.94
Beam search (width =5)	**50.05**	**46.39**

Table 5: Comparison among different sample methods

Beam search can further improve the performance. The beam width 5 benefits the performance most. Beam width larger than 5 can be slow and the improvement is flatten out.

Captioning Examples

Input Video:

Output:
a dolphin is swimming in the water.

Even though it is hard to "see" what's under the water, the model is able to guess it is likely to be a dolphin swimming.

Input Video:

Output:
a dog is playing with a toy.

The model is able to tell there are multiple important objects (a dog and a toy) and their relationship (playing with).

Input Video:

Output:
a woman is cleaning a tub.

The model is able to capture different human motions like cleaning and climbing which is very impressive.

Input Video:

Output:
A man is climbing a rock wall.

Reference

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020, 2021.