Video Paragraph Captioning as a Text Summarization Task

Hui Liu, Xiaojun Wan
Wangxuan Institute of Computer Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
{xinkeliuhui, wanxiaojun}@pku.edu.cn

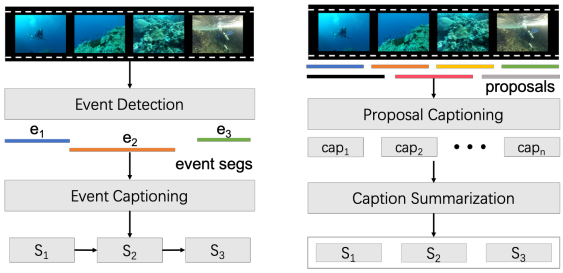
Abstract

Video paragraph captioning aims to generate a set of coherent sentences to describe a video that contains several events. Most previous methods simplify this task by using ground-truth event segments. In this work, we propose a novel framework by taking this task as a text summarization task. We first generate lots of sentence-level captions focusing on different video clips and then summarize these captions to obtain the final paragraph caption. Our method does not depend on ground-truth event segments. Experiments on two popular datasets ActivityNet Captions and YouCookII demonstrate the advantages of our new framework. On the ActivityNet dataset, our method even outperforms some previous methods using ground-truth event segment labels.

1 Introduction

Video captioning, the task of describing the content of a video in natural language, is a popular task both in computer vision and natural language processing. In the beginning, researchers try to generate sentence-level captions for short video clips (Venugopalan et al., 2015). Krishna et al. (2017) propose the task of dense video captioning. The system needs to detect event segments first and then generate captions. Park et al. (2019) propose the task of video paragraph captioning: they use ground-truth event segments and focus on generating coherent paragraphs. Lei et al. (2020) follow the task setting and propose a recurrent transformer model that can generate more coherent and less repetitive paragraphs. Considering the ground-truth event segments are often unavailable in practice, our goal is to generate paragraph captions without ground-truth segments.

The conventional framework of video paragraph captioning is shown in Figure 1a. Given an untrimmed video, an Event Detection module outputs a set of non-redundant event segments. The Event Captioning module generates captions for these segments. The works of (Park et al., 2019; Zhou et al., 2019; Lei et al., 2020) use ground-truth event segments and focus on the Event Captioning module. Zhou et al. (2019) use extra human-annotated bounding boxes as supervision. (Sah et al., 2017; Zhou et al., 2018; Mun et al., 2019) use predicted event segments and generate captions based on them. Sah et al. (2017) also summarizes these captions to generate a paragraph. The above methods heavily depend on accurate event segments. According to previous works (Zhou et al., 2018; Mun et al., 2019), the performance of the Event Detection module is not so good, making it a performance bottleneck. To tackle this problem, we propose a novel framework VPCSum as shown in Figure 1b. For a given video, we first extract dense event segment candidates (we call proposals), and a Proposal Captioning module is used to generate proposal captions. Then we treat video paragraph captioning as a text summarization task to obtain the final summary (paragraph caption).



(a) Conventional Framework (b) Our Framework

Figure 1: Comparison between conventional framework and ours.

In this work, we only consider extractive summarization, where the paragraph caption is composed by selecting from proposal captions. We conduct experiments on two popular datasets ActivityNet

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 55–60
August 1–6, 2021.

Captions and YouCookII. The results demonstrate the advantages of our framework. On the ActivityNet Captions dataset, our method even outperforms some previous methods using ground-truth event segment labels.

2 Our VPCSum Method

As illustrated in Figure 1b, our framework has three modules. **Proposal Extraction**: it extracts dense proposals for a video; **Proposal Captioning**: it generates captions for extracted proposals; **Caption Summarization**: it summarizes the generated proposal captions to obtain the video paragraph caption. We will introduce each module next.

2.1 Proposal Extraction

For proposal extraction, we use the BMN model (Lin et al., 2019), a popular model for temporal action proposal generation. It can extract complete and accurate proposals. We extract the top 100 proposals for each video.

2.2 Proposal Captioning

For proposal captioning, we choose the TSRM-RNN model (Wang et al., 2020) for ActivityNet Captions and VTransformer model (Lei et al., 2020) for YouCookII according to proposal captioning performance. We believe that if we choose a better sentence-level captioning model, the performance can be further improved.

2.3 Caption Summarization

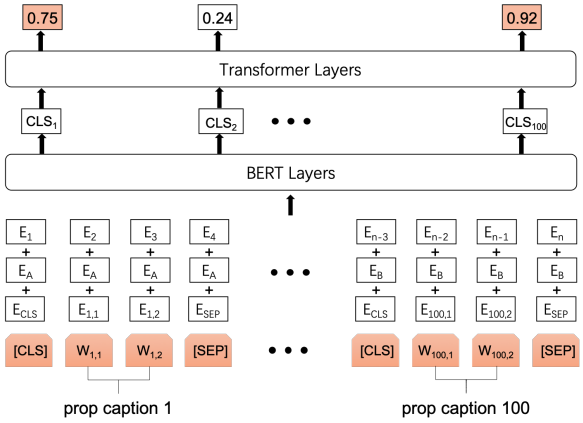


Figure 2: Architecture of the caption summarization model.

The caption summarization module summarizes proposal captions to generate the final video paragraph caption. In this work, we focus on extractive summarization. The architecture of our summarization model is illustrated in Figure 2. We first sort the proposal captions according to the proposal start time and add special [CLS] and [SEP] tokens to the beginning and end of each caption. We use the summation of token embeddings, segment embeddings, and position embeddings to represent each word. The input representations are fed into a pre-trained BERT model (Devlin et al., 2018), after which we obtain the contextual token representations. We use the contextual vectors of [CLS]s to represent each caption and feed them into stacked transformer layers (Vaswani et al., 2017). We use a sigmoid layer to compute the score of each caption:

$$x_i = \sigma(W h_i^L + b) \quad (1)$$

where W and b are trainable parameters, h_i^L is the vector for caption i from the top transformer layer.

For extractive summarization, we need to annotate each sentence according to the gold summary as our training target. Many researchers use a greedy algorithm (Nallapati et al., 2016), sentences are selected one by one to maximize the ROUGE score against the gold summary. The selected sentences are labeled 1 while others are labeled 0 (hard-label). In our task, we find a more effective soft-label annotation method. We label caption c_i with the max ROUGE score against gold captions and use binary cross-entropy as our loss function:

$$y_i = \max_{g_j \in gold} ROUGE(c_i, g_j) \quad (2)$$

$$\mathcal{L} = -\sum_i (y_i \log x_i + (1 - y_i) \log(1 - x_i)) \quad (3)$$

where g_j is the j-th gold caption.

2.4 Leverage Visual Information

The above caption summarization module assigns each proposal caption a predicted score, indicating how likely it appears in the final paragraph caption. The predicted score only depends on text information. To leverage visual information, we need a "visual summarization" module, which gives a visually weighting score to each proposal. The ESGN model (Mun et al., 2019) seems a good choice for us. It uses a pointer network to select events from proposals and assigns a visually weighting score for each proposal. We use this model to compute the visually weighting score.

Now we can extract the final paragraph caption. The final score of each proposal caption is a

weighted sum of the textually weighting score s_{txt} and the visually weighting score s_{vis}:

$$score(i) = s_{txt,i} + \lambda s_{vis,i} \quad (4)$$

where λ is a hyper-parameter tuned on validation set. We select captions according to $score(i)$ and use Trigram Blocking to reduce redundancy, as in Liu and Lapata (2019).

3 Experiments

3.1 Datasets

We conduct experiments on ActivityNet Captions (Krishna et al., 2017) and YouCookII (Zhou et al., 2017). ActivityNet Captions contains 10,009 videos in train set, 4,917 videos in val set. Each video has 3.65 event segments on average. Following (Lei et al., 2020), the original val set is split into ae-val with 2,460 videos for validation and ae-test with 2,457 videos for test. YouCookII contains 1,333 videos in train set, 457 videos in val set. Each video has 7.70 event segments on average.

3.2 Evaluation Metrics

Following (Lei et al., 2020; Park et al., 2019), we evaluate the captioning performance at paragraph level. We report standard caption metrics, including BLEU@4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), CIDEr (Vedantam et al., 2015). We also evaluate repetition using R@4 (Xiong et al., 2018). We use the scripts provided by (Lei et al., 2020) for evaluation[1].

3.3 Implementation Details

For video preprocessing, we use appearance and optical flow features provided by Zhou et al. (2018). For BMN model and captioning models, we use the same hyperparameters suggested by the authors. For ESGN model, we use a transformer encoder instead of an RNN encoder, with hidden size set to 512, number of heads set to 8, number of layers set to 3. For our caption summarization model, we use the base BERT model, 2 stacked transformer layers with hidden size set to 768, number of heads set to 8. We set max input length to 1,700, batch size to 10, λ to 1 for ActivityNet Captions and max input length to 1,000, batch size to 1, λ to 1 for YouCookII. Warmup steps are set to step num of 1 epoch. We use Adam optimizer with an initial learning rate of $6e-4$.

[1] https://github.com/jayleicn/recurrent-transformer

3.4 Baselines and Results

We compare our **VPCSum** model with the following baselines. **Soft-NMS**: it uses Soft-NMS (Bodla et al., 2017) to select event segments from BMN proposals, and uses the proposal captioning model to generate captions; **ESGN**: similar to Soft-NMS, but it uses ESGN model (Mun et al., 2019) to select event segments from BMN proposals; **V-Trans**: a Vanilla Transformer model, proposed by (Zhou et al., 2018); **Trans-XL**: a Transformer-XL model, proposed by (Lei et al., 2020); **MART**: a recurrent transformer model (Lei et al., 2020); **COOT**: it uses pretrained features to train MART model (Ging et al., 2020). Originally, the last four models deal with ground-truth event segments. For fair comparison, we also test them with predicted event segments generated by ESGN model [2].

Models	B@4	M	C	R@4↓
Soft-NMS	10.33	14.93	22.58	10.17
ESGN	10.38	15.74	21.85	6.51
V-Trans	9.89	15.11	20.95	7.04
Trans-XL	10.36	14.89	20.73	7.45
MART	10.13	14.94	20.16	6.09
COOT	9.85	14.67	21.83	7.15
VPCSum	**10.89**	**15.84**	**24.33**	**1.54**
V-trans*	9.31	15.54	21.33	7.45
Trans-XL*	10.25	14.91	21.71	8.79
MART*	9.78	15.57	22.16	**5.44**
COOT*	**10.85**	**15.99**	**28.19**	6.64

Table 1: Comparison with baselines on ActivityNet Captions ae-test split. * means the model uses ground-truth event segments. We report BLEU@4 (B@4), METEOR (M), CIDEr (C), Repetition (R@4).

Tables 1 and 2 show the results on ActivityNet Captions and YouCookII. We can observe that on the ActivityNet Captions, our model **VPCSum** within the new framework can generate better paragraph captions with higher Bleu@4, METEOR, and CIDEr and lower repetition score R@4, even outperforming V-trans*, Trans-XL*, MART* models using ground-truth event segments on every metric. On the YouCookII dataset, our model outperforms the models in the same setting but is inferior to the models using ground-truth segments. This may be because YouCookII has more segments

[2] We use the codes and pretrained models provided by the authors and only replace ground-truth event segments with ESGN predicted event segments.

Models	B@4	M	C	R@4↓
Soft-NMS	5.58	13.67	18.18	4.94
ESGN	5.36	13.37	17.01	2.82
V-Trans	5.35	13.37	16.88	2.85
Trans-XL	4.78	12.67	14.24	3.20
MART	5.61	13.44	16.56	4.63
COOT	5.96	14.21	19.67	5.99
VPCSum	**6.14**	**15.11**	**23.92**	**0.65**
V-trans*	7.62	15.65	32.26	7.83
Trans-XL*	6.56	14.76	26.35	6.30
MART*	8.00	15.90	35.74	**4.39**
COOT*	**9.44**	**18.17**	**46.06**	6.30

Table 2: Comparison with baselines on YouCookII val split.

(7.70 vs 3.65) than ActivityNet Captions.

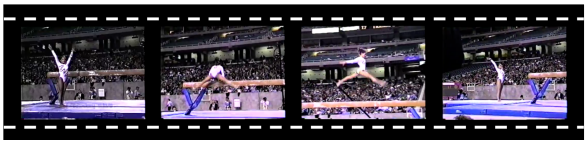
3.5 Ablation Study

Table 3 shows the ablation study on ActivityNet Captions. Compared to our full model (Full), the traditional extractive summarization annotation method (Hard-label) is not suitable for our task. If we set λ in Eq.(4) to 0 (w/o vis), the model loses useful visual information and performs not well. If we remove Trigram Blocking (w/o tri-blk), the performance also degrades and repetition becomes a problem (R@4 increases to 7.91). To verify the role of pretrained BERT model, we retrain our VPCSum without BERT pretrained weights (w/o pretrain). We can see that BERT pretrained weights are not the major factor to the final performance. We also replace our summarization model with unsupervised methods LexRank (Erkan and Radev, 2004) and LSA(Steinberger and Jezek, 2004). The results show that simple unsupervised summarization methods cannot handle our data well and supervised training is necessary.

Models	B@4	M	C	R@4↓
Full	10.89	15.84	24.33	1.54
Hard-label	10.29	14.99	21.71	1.19
w/o vis	10.68	15.78	23.34	1.36
w/o tri-blk	10.46	15.61	21.40	7.91
w/o pretrain	10.84	15.81	24.00	1.55
LexRank	7.78	13.65	14.19	26.51
LSA	7.24	14.48	12.43	28.14

Table 3: Model ablation study on ActivityNet Captions ae-test split.

3.6 Qualitative Results



Ground Truth: A girl jumps onto a balance beam. She does a gymnastics routine on the balance beam. She does a flip off the balance beam and lands on a mat.
MART: A gymnast is seen standing ready with her arms up and leads into her performing a gymnastics routine. She continues performing several flips and tricks and tricks and ends with her jumping down and walking away. She continues her routine and ends with her jumping down and jumping down and walking away.
MART*: A gymnast is seen standing before a beam and begins performing a gymnastics routine. The girl then performs a routine on the beam and ends with her jumping down and jumping down and jumping. The girl jumps off the beam and lands on the mat and jumps off the beam.
VPCSum: A gymnast is seen standing ready with her arms up and begins to do a routine. She does a gymnastics routine on the beam. She dismounts and lands on the mat.

Figure 3: An example from ActivityNet Captions.

We show an example in Figure 3 with paragraph captions generated by MART, MART* and our VPCSum model. Compared to other models, our model can generate more clear and correct sentences with less redundancy. The generated paragraph of our model can better describe the process of the whole event.

3.7 Human Evaluation

	Ours	MART	Ours	MART*
rel.	56.0%†	44.0%†	52.7%	47.3%
div.	56.7%†	43.3%†	56.7%†	43.3%†

Table 4: Human evaluation results. Statistically significant differences ($p < 0.05$) are marked with †.

We also conduct a human evaluation on randomly sampled 50 videos from the ActivityNet Captions val set. The annotators are asked to choose the better caption from two models in two aspects: **relevance** (how related is the caption to the video content) and **diversity** (how diverse is the generated text). We compare our VPCSum model with MART and MART* respectively. We have 17 college students as our annotators. Each video is judged by 3 annotators. We show the results of the pairwise experiments in Table 4. Our VPCSum model performs better in relevance and diversity,

and more people choose the caption of our model as the better one.

4 Conclusion

In this work, we view the task of video paragraph captioning as a text summarization task and propose a novel framework VPCSum. It allows us to use text summarization techniques to handle this challenging task. Experimental results on two popular datasets show the advantages of our model. In the future, we will explore using abstractive summarization methods to generate better video paragraph captions.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036), MSRA Collaboration Research Project (FY20-Research-Sponsorship-266) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. 2017. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. Coot: Cooperative hierarchical transformer for video-text representation learning. *arXiv preprint arXiv:2011.00597*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. 2020. Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. *arXiv preprint arXiv:2005.05402*.

Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. 2019. Streamlined dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6588–6597.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *arXiv preprint arXiv:1611.04230*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. 2019. Adversarial inference for multi-sentence video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6598–6608.

Shagan Sah, Sourabh Kulhare, Allison Gray, Subhashini Venugopalan, Emily Prud'Hommeaux, and Raymond Ptucha. 2017. Semantic text summarization of long videos. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 989–997. IEEE.

Josef Steinberger and Karel Jezek. 2004. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4:93–100.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542.

Teng Wang, Huicheng Zheng, and Mingjing Yu. 2020. Dense-captioning events in videos: Sysu submission to activitynet challenge 2020. *arXiv preprint arXiv:2006.11693*.

Yilei Xiong, Bo Dai, and Dahua Lin. 2018. Move forward and tell: A progressive generator of video descriptions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 468–483.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. 2019. Grounded video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6578–6587.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2017. Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv:1703.09788*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.