Supplemental Materil to CLIP-It! Language-Guided Video Summarization

Medhini Narasimhan Anna Rohrbach Trevor Darrell
University of California, Berkeley
{medhini, anna.rohrbach, trevordarrell}@berkeley.edu
https://medhini.github.io/clip_it

This section is organised as follows:

1. Details about the Datasets
2. Implementation Details
3. Additional Results
4. Limitations

Additionally we include the following videos in clipit_results.mp4:

1. Summary videos for Fig.1 in the main paper. Note that this result is from applying our method in the wild, as the video was not a part of any dataset and was downloaded from YouTube.[1] This shows that our method can generalize to out-of-distribution data.
2. Result from TVSum dataset
3. Result from SumMe dataset
4. Result from QFVS dataset

1 Details about the Datasets

Note. All the datasets - YouTube (1), Open Video Project (OVP) dataset (5), TVSum (12), SumMe (2), and QFVS (10) were collected by the creators (cited) and consent for any personally identifiable information (PII) was ascertained by the authors where necessary.

TVSum (12) consists of 50 videos pertaining to 10 categories (how to videos, news, documentary, etc) with 5 videos from each category, typically 1-5 minutes in length. SumMe (2) consists of 25 videos capturing multiple events such as cooking and sports, and the lengths of the videos vary from 1 to 6 minutes. In addition to training on each dataset independently, we follow prior work and augment training data with 39 videos from the YouTube dataset (1) and 50 videos from the Open Video Project (OVP) dataset (5). YouTube dataset consists of news, sports and cartoon videos. OVP dataset consists of multiple different genres including documentary videos. These datasets are diverse in nature and come with different types of annotations, frame-level scores for TVSum and shot-level scores for SumMe. They are integrated to create the ground-truth using the procedure in (14). The UT Egocentric dataset consists of 4 videos captured from head-mounted cameras. Each video is about 3-5 hours long, captured in a natural, uncontrolled setting and contains a diverse set of events. The QFVS dataset (11) provides ground-truth generic summaries for these 4 videos. The summaries were constructed by dividing the video into shots and asking 3 users to select the relevant shots. The final ground-truth is an average of annotations from all users.

2 Implementation Details

Language-Guided Multi-head Attention. We use multi-head attention with 4 heads. We pass the Image Encoding as the Query and the Text Encoding as the Key and Value.

[1] https://www.youtube.com/watch?v=yN0xiSMnUww

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Frame-Scoring Transformer. We use a Transformer with 8 heads, 6 encoder layers and 6 decoder layers. The length of the sequence passed as input to the Transformer was heuristically chosen as 256.

Image Encoding. We encode the image using the CLIP (8) Image encoder to obtain image encoding $f_{img}(F) \in \mathbb{R}^{512}$.

Text Encoding. For query-focused video summaries, we encode the query using the CLIP Text encoder to obtain text embedding $f_{text}(C) \in \mathbb{R}^{512}$. For generic video summaries, we first generate dense video description using BMT (3) by sampling frames from the input video at 2 fps. For a 2-3 min video BMT generates 10-15 sentences. Next, we uniformly sample 7 sentences from the dense description corresponding to different video segments over time. Each sentence is then encoded using CLIP text encoder and the 7 embeddings are concatenated to obtain a feature vector. This is passed through a linear layer to obtain the input text embedding. Heuristically, we found that sampling 7 captions worked best for TVSum and SumMe datasets where the average duration of the videos is 2 mins. For generic summarization on the QFVS dataset (day long videos) reported above, the frames are extracted at 2 FPS and pass this through the BMT pipeline. This generates roughly 20 sentences and we then sampled 15 captions for each video since the videos are significantly longer.

Next, we uniformly sample M captions from the dense description corresponding to different video segments over time. Each caption is then encoded using CLIP Text encoder and the M embeddings are concatenated to obtain a feature vector in R^{MX512}. This is passed through a linear layer to obtain $f_{text}(C) \in \mathbb{R}^{512}$. We find that $M = 7$ works best.

Table 1: Kendall's τ (4) and Spearman's ρ (16) correlation coefficients computed on the TVSum benchmark (12).

Method	Kendall's τ	Spearman's ρ
Zhang et al. (14)	0.042	0.055
Zhou et al. (15)	0.020	0.026
Park et al.(SumGraph) (7)	0.094	0.138
CLIP-It	**0.108**	**0.147**
Human	0.177	0.204

Training. Note that the caption generator, image and text encoders are kept fixed. The Language-Guided Multi-Headed Attention network and the Frame-Scoring Transformer are trained using Adam optimizer and a learning rate of 1e-4 and weight decay of 0.001.

Computational Resources. For each dataset and data setting, we train our method for 20 epochs with a batch size of 100 which takes about 2-3 hours on 5 NVIDIA RTX 2080 GPUs.

Frame Scores to Shot Scores. For Generic Video Summarization, different datasets provide ground-truth annotations in different formats. Following (13; 14), we obtain a single set of ground-truth keyframes (small subset of isolated frames) for each video. If a frame is selected to be a part of the summary, it is labeled 1, otherwise 0. The model is trained using keyframe annotations but evaluated on keyshots (interval-based subset of frames). For fair comparison, we follow (13; 14; 9) to convert the keyframes to keyshots.

For Query-Focused Video Summarization, the video is divided into shots of 5 seconds each (10). Ground-truth annotations are available for each shot. While prior work predicts a single score per shot, we predict scores for each frame in the shot. In order to combine the scores for all frames in a shot we use two strategies: taking the max and taking the average. We found that taking the average of scores assigned to all frames in a shot to determine the shot score works best.

3 Additional Results

We also follow Otani *et al.* (6) and report results on rank based metrics, Kendall's τ (4) and Spearman's ρ (16) correlation coefficients in the Tab. 1 for TVSum. They are computed by first ranking the frames in the video based on the predicted scores and the ground-truth scores and then comparing the two rankings. The correlation scores are computed by averaging over the individual results. We outperform all the baselines on these metrics as well.

Table 2: F1 scores of **CLIP-It** for different loss ablations.

Method	SumMe			TVSum		
	Standard	Augment	Transfer	Standard	Augment	Transfer
$\mathcal{L}_c$	49.1	52.6	46.2	60.2	61.7	57.3
$\mathcal{L}_c + \mathcal{L}_r$	53.0	55.4	50.3	64.5	66.5	63.4
$\mathcal{L}_c + \mathcal{L}_d$	53.7	55.8	50.8	65.4	67.6	64.3
$\mathcal{L}_{unsup} = \mathcal{L}_d + \mathcal{L}_r$	52.5	54.7	50.0	63.0	65.7	62.8
$\mathcal{L}_{sup} = \mathcal{L}_c + \mathcal{L}_d + \mathcal{L}_r$	54.2	56.4	51.9	66.3	69.0	65.5

Table 3: Ablating the Cross-Modal Attention module.

Method	SumMe			TVSum		
	Standard	Augment	Transfer	Standard	Augment	Transfer
CLIP-It (MLP)	50.6	51.08	48.1	63.0	65.8	61.4
CLIP-It (Cross-Modal Attn)	54.2	56.4	51.9	66.3	69.0	65.5

In Tab. 2, we ablate the different loss functions described in Sec. 3 of the main paper and report results on TVSum and SumMe datasets. $\mathcal{L}_c$ is the Classification loss, $\mathcal{L}_r$ is the Reconstruction loss, and $\mathcal{L}_d$ is the Diversity loss. Results shown are for the our full model, CLIP-It. Parameters α, β, and λ described in Sec. 3 are chosen heuristically and are set to 0.5, 0.3 and 0.2 respectively.

As we see, the Classification loss alone yields the lowest F1 scores. Adding the Reconstruction or Diversity losses improves performance. However, in the unsupervised setting, as the ground truth annotations cannot be used, we remove the Classification loss. This causes a slight drop in performance. Our method works best in the supervised setting, when all three losses are combined.

4 Limitations

As described, we use large scale language models for video captioning (3) and feature extraction (CLIP (8)) which may have encoded some inappropriate biases that could propagate to our model. In particular, as CLIP was trained on 400M image-caption pairs sourced from the Web, we can not rule out the presence of biases or stereotypes which may propagate into how the video frames are scored within our method.

References

[1] De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., de Albuquerque Araújo, A.: Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Patt. Rec. Letters (2011) 1

[2] Gygli, M., Grabner, H., Riemenschneider, H., Gool, L.V.: Creating summaries from user videos. European Conference on Computer Vision (ECCV) (2014) 1

[3] Iashin, V., Rahtu, E.: A better use of audio-visual cues: Dense video captioning with bi-modal transformer. British Machine Vision Conference (BMVC) (2020) 2, 3

[4] Kendall, M.G.: The treatment of ties in ranking problems. Biometrika **33**(3), 239–251 (1945) 2

[5] Open video project. https://open-video.org/ 1

[6] Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J.: Rethinking the evaluation of video summaries. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2

[7] Park, J., Lee, J., Kim, I.J., Sohn, K.: Sumgraph: Video summarization via recursive graph modeling (2020) 2

[8] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021) 2, 3

[9] Rochan, M., Ye, L., Wang, Y.: Video summarization using fully convolutional sequence networks. European Conference on Computer Vision (ECCV) (2018) 2

[10] Sharghi, A., Gong, B., Shah, M.: Query-focused extractive video summarization (2016) 1, 2

[11] Sharghi, A., Laurel, J.S., Gong, B.: Query-focused video summarization: Dataset, evaluation, and a memory network based approach (2017) 1

[12] Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: Tvsum: Summarizing web videos using titles. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 1, 2

[13] Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Summary transfer: Examplar-based subset selection for video summarization. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2

[14] Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. European Conference on Computer Vision (ECCV) (2016) 1, 2

[15] Zhou, K., Qiao, Y., Xiang, T.: Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. The Association for the Advancement of Artificial Intelligence Conference (AAAI) (2018) 2

[16] Zwillinger, D., Kokoska, S.: Crc standard probability and statistics tables and formulae. CRC Press (1999) 2