Unsupervised Video Summarization Experimental Results

Minh-Nam Tran (20125039) Viet-Nhat Thai (20125014)

December 16, 2023

1 Introduction

2 Related Works

In the 2020s, the volume of video has been observed to an exponential increase, especially on social media platforms. It leads to the issues of storing, managing, retrieving, and sharing videos. Therefore, an emerging need for fast, efficient video summarization has been raised.

Video summarization is the task of generating a small sequence of frames that contain the most critical and relevant data of the video clip in a synopsis or condensed form [1, 2]. Based on the type of training dataset, video summarization techniques are categorized into supervised and unsupervised. The supervised method requires the annotations of keyframes in the training video samples and tries to minimize the loss of the summarizer concerning the labels [3, 4, 5]. On the other hand, the unsupervised method uses unlabeled data, which does not require much human effort to create training data. Since a wide range of domains makes providing reliable and accurate large amounts of labeled datasets impossible, the unsupervised method is considered an essential direction in video summarization.

Before 2015, the scientists focused on hand-crafted patterns to select the essential frames through the scores of the heuristic functions [6, 7, 8, 9]. Since then, the attention to deep-learning-based methods has been increased with the adaptation of neural networks; many works use the convolutional neural network to handle generic feature extraction and the recurrent network to capture long-range dependency between the video frames [5]. To make the networks learn from unsupervised datasets, two training frameworks are effectively facilitated, including reinforcement learning [10, 11] and generative adversarial network [12, 13, 14].

In 2017, B. Mahasseni et al. proposed SUM-GAN [12], a framework that optimizes the deep summarizer under the generative adversarial framework [15]. The framework contains two components: summarizer and GAN. The summarizer is the autoencoder [16] LSTM network focusing on selecting keyframes, encoding, and decoding the selected frames for video reconstruction in the GAN component. The discriminator in the GAN component is trained to recognize the original and synthesized videos. In this technique, the summarizer learns to select the most essential frames to provide sufficient information for the decoder to reconstruct the whole video, i.e., trained to confuse the discriminator. Inspired by the novel idea of B. Mahasseni et al. [12], Y. Jung proposed Chunk and Stride Network (CSNet) for discriminative feature learning [13]. The input video stream is split into two flows, chunking and striding, then passed through the LSTM cell and merged back for calculating the result.

Besides the GAN series for unsupervised learning, several studies employ the reinforcement learning algorithm. K. Zhou et al. proposed DSN [10], a deep summarization network built on CNN and bidirectional LSTM. The technique considered the input video, which is basically a sequence of frames, as a consecutive collection of decision problem and modeled that each frame may be chosen or not to appear in the summary. The DSN is trained under the reinforcement learning algorithm with the help of the reward function to determine the diversity level and the meaningful of the summaries. Another variation of recurrent neural network,

independently recurrent neural network (IndRNN) [17], is used in the method proposed by G. Yaliniz et al. The authors constructed a special reward function to support the reinforcement learning process, aiming to achieve greater consistency, variety and representativeness in the output summaries. In short, the unsupervised video summarizing algorithms trained under the reinforcement learning framework focusing on designing the reward function that boosts the model performance.

When the models are built with recurrent neural networks, the problems are the ability to capture long-range dependency of the RNN units and the slow computation since RNN cells are unable to handle data parallelly. Therefore, the attention mechanism appears to solve the two problems. E. Apostolidis et al. use Luong's attention [18] between the encoder and the decoder of the variational autoencoder [16]. Besides the Luong's attention, other studies employ the self-attention mechanism [19, 20]. X. He et al. propose attentive conditional GAN [21] with three modules: feature descriptor, generator and discriminator. All modules are built with self-attention modules, but the generator and the discriminator also use bidirectional LSTM, which causes a bottleneck for this network. Similar to X. He, G. Liang et al. replaced the recurrent neural network components with the self-attention module but not completely [14]. Meanwhile, G.Liang introduced CAAN [14], which uses the self-attention in the generator to find the relationship between features from a fully convolutional sequence network. However, the disavantage of this method is that the discriminator still uses the LSTM network, which causes bottleneck effect.

Since using LSTM cells inside the state-of-the-art frameworks slows down training and inference procedures, the proposed method aimed to replace those LSTM cells completed by the self-attention module. More information is described in Section 3.

3 Proposed Method

In this section, MHA-SUM-GAN method is introduced by leveraging the Transformer block with the multi-head attention module to extract the relationship between video frames. The proposed method is evaluated on two benchmark datasets, SumMe and TVSum.

3.1 Datasets

This work uses two datasets to evaluate and compare with the state-of-the-art benchmark. The SumMe dataset [22] is a collection of 25 videos, each annotated by at least 15 human summaries. TVSum dataset [9] is a set of videos crawled from YouTube. The benchmark contains 50 videos covering various genres: news, instruction, documentary, vlog, and egocentric. Every video annotation is the shot-level importance score gained by crowd-sourcing.

3.2 MHA-SUM-GAN

The pipeline contains two main components, the summarizer and the GAN. The summarizer includes the Selector, the Encoder, and the Decoder, while the GAN framework has the Decoder and the Discriminator. In this method, all the modules except the CNN are built on top of the Multi-Head Attention [19] combined with a specific Linear head to map the dataflow into the correct dimensional space. Using the Multi-Head Attention, the problems of capturing long-term dependencies and bottlenecks when computing have been solved.

The architecture of MHA-SUM-GAN is visualized in Figure 1. The subscript t denotes the $t-th$ sample within N samples in the dataset ($t \in [1, N]$.

First, the features $x_t \in R^{L \times d}$ extracted from the video frame-by-frame via GoogLeNet [23] (L for the number of frames in the video, d for the hidden size, or the number of features). This step is done in the preprocessing stage.

Then, the captured features x_t is passed through the Attentive Selector to obtain the frame important scores $s_t \in R^{L \times 1}$, also known as the probability of a frame appearing in the summary. Then, input features x_t are scaled with the corresponding important scores s_t to

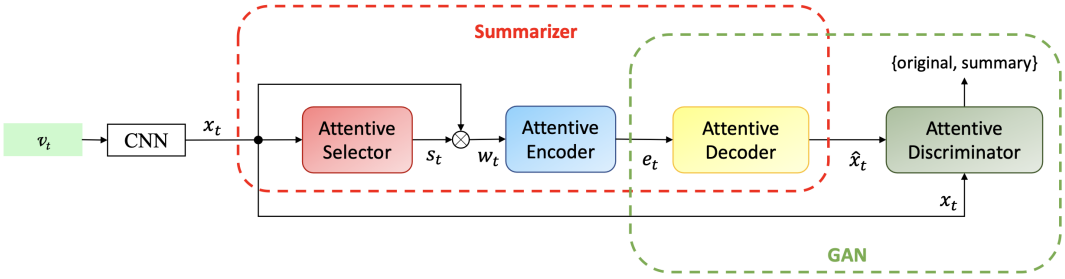

Figure 1: Architecture of MHA-SUM-GAN.

obtain the weighted features w_t. After that, the Attentive Encoder receives w_t and encodes it into $e_t \in R$.

The following stages happen in the GAN framework. The encoded features e_t are then passed through the Attentive Decoder to reconstruct the feature sequences $\hat{x}_t \in R^{L \times d}$. Next, the revamped sequence $\hat{x}_t$ and x_t are considered as training samples of the Attentive Discriminator, where it is trained to recognize the real (original) and fake (generated) samples.

The loss function for this method is similar to the method of G. Liang et al. [14], where there are 3 smaller components: adversarial loss, reconstruction loss and sparsity loss. The adversarial loss is used to make the decoded weighted features w_t similar to the original features x_t. The reconstruction loss tries to minimize the difference between the generated features $\hat{x}_t$ and the original ones x_t. Finally, the sparsity loss emphasize the model to focus on less frames in the selector.

4 Experimental Results

In this section, the experiments with the proposed architecture on the TVSum datasets are conducted. Besides testing the new architecture, the comparison with the baseline architecture, SUM-GAN-AAE.[12], are also conducted with the same hyperparameter setting.

4.1 Datasets

The performance of the models is calculated on the SumMe [22] and TVSum [9] datasets.

SumMe contains 25 videos, where each of which ranging in duration from 1 to 6 minutes. These videos capture various events from both a first-person and third-person perspective. Each video clip has been annotated by 15 to 18 users using key-fragments, resulting in many fragment-level user summaries. Additionally, each ground-truth summary is given in the form of frame-level importance scores, which are computed by averaging the key-fragment user summaries for each frame.

The TVSum dataset is a collection of 50 movies. Each movie has the duration ranging from 1 to 5 minutes. The dataset covers 10 different categories from the TRECVid MED dataset. Each movie has been annotated by 20 users using frame-level relevance scores, and a single ground-truth summary calculated by averaging all the scores.

4.2 Evaluation Metric

To compare the generated summary and the groundtruth sequence, the key-fragment-based evaluation protocol [5] and F1 score (also known as F-score) are used.

The F-Score, also known as F1 score, is a metric commonly used in binary classification to evaluate the performance of a model. It is the harmonic mean of precision and recall ((see Equation 1). In this framework, the video summarization problem is shifted to a binary classification task where the model needs to decide if each frame is included in the summary or not.

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \text{ where } P = \frac{TP}{TP + FP} \text{ and } R = \frac{TP}{TP + FN} \tag{1}$$

4.3 Hyperparameter Settings

Model name	#Layers	#Hidden	#Heads	#FeedForward	Model Size
Attentive-SUM-GAN$_{\text{small}}$	2	256	4	512	4.480M
SUM-GAN-AAE$_{\text{small}}$	2	256	-	-	6.510M
Attentive-SUM-GAN$_{\text{base}}$	2	512	4	1024	17.348M
SUM-GAN-AAE$_{\text{base}}$	2	512	-	-	25.472M
Attentive-SUM-GAN$_{\text{large}}$	4	512	4	1024	34.170M
SUM-GAN-AAE$_{\text{large}}$	4	512	-	-	51.203M

Table 1: Model configuration for SUM-GAN-AAE and Attentive-SUM-GAN.

For training, the hyperparameters are listed as follows:

- Adam optimizer is used with selector and generator learning rate of $1e-4$, discriminator learning rate of $1e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and regularization factor of 0.15.
- Each training session run the models through 100 epochs, and the gradient is clipped to have maximum value of 5.0

The model configuration is shown in Table 1. With the same number of layers and hidden size, the SUM-GAN-AAE model always has more number of trainable parameters than the Attentive-SUM-GAN model.

4.4 Training Time

The following training time is recored of the models for a single session over two datasets in the experiments. The statistics are included in Table 2.

Model	SumMe	TVSum
Attentive-SUM-GAN$_{\text{small}}$	00:05:14	00:05:05
SUM-GAN-AAE$_{\text{small}}$	01:29:11	01:23:57

Table 2: Training time of Attentive-SUM-GAN and SUM-GAN-AAE. The reported time is in format HH:MM:SS.

From Table 2, it is noticeable that the training time of Attentive-SUM-GAN is 18 times faster than the SUM-GAN-AAE method, which is basically implemented on LSTM cells. By using Transformer architecture, the training time greatly reduced.

4.5 Performance Benchmarking

4.5.1 Small Configuration Models

The evaluation results are presented in Table 3, indicating F-Score percentages for Attentive-SUM-GANsmall and SUM-GAN-AAEsmall across five splits (0 to 4), as well as the average performance.

Split	0	1	2	3	4	Average
Attentive-SUM-GAN$_{\text{small}}$	62.07	61.94	65.44	63.83	64.80	63.62
SUM-GAN-AAE$_{\text{small}}$	61.84	64.24	63.03	61.29	64.15	62.91

Table 3: F-Score (%) of the baseline and proposed method in small configuration on TVSum .

For Attentive-SUM-GAN$_{\text{small}}$, the F-Score values range from 61.94% to 65.44%, with an overall average of 63.62%. Meanwhile, SUM-GAN-AAE$_{\text{small}}$ demonstrates F-Scores ranging from 61.29% to 64.24%, resulting in an average of 62.91%.

Overall, with small configuration, Attentive-SUM-GAN outperforms SUM-GAN-AEE in both average performance and training time.

4.5.2 Ablation Studies

References

[1] V. Tiwari and C. Bhatnagar, "A survey of recent work on video summarization: approaches and techniques," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 27187–27221, 2021.

[2] M. Basavarajaiah and P. Sharma, "Survey of compressed domain video summarization techniques," *ACM Comput. Surv.*, vol. 52, oct 2019.

[3] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[4] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1059–1067, 2016.

[5] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pp. 766–782, Springer, 2016.

[6] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Automatic video summarization by graph modeling," in *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 104–109 vol.1, 2003.

[7] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, "Large-scale video summarization using web-image priors," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2698–2705, 2013.

[8] B. Zhao and E. P. Xing, "Quasi real-time summarization for consumer videos," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2513–2520, 2014.

[9] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5179–5187, 2015.

[10] K. Zhou, Y. Qiao, and T. Xiang, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018.

[11] S.-S. Zang, H. Yu, Y. Song, and R. Zeng, "Unsupervised video summarization using deep non-local video summarization networks," *Neurocomputing*, vol. 519, pp. 26–35, 2023.

[12] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2982–2991, 2017.

[13] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, "Discriminative feature learning for unsupervised video summarization," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19, AAAI Press, 2019.

[14] G. Liang, Y. Lv, S. Li, S. Zhang, and Y. Zhang, "Unsupervised video summarization with a convolutional attentive adversarial network," *arXiv preprint arXiv:2105.11131*, 2021.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[17] G. Yaliniz and N. Ikizler-Cinbis, "Using independently recurrent networks for reinforcement learning based unsupervised video summarization," *Multimedia Tools and Applications*, vol. 80, pp. 17827–17847, 2021.

[18] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (L. Màrquez, C. Callison-Burch, and J. Su, eds.), (Lisbon, Portugal), pp. 1412–1421, Association for Computational Linguistics, Sept. 2015.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, Ł. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*, pp. 7354–7363, PMLR, 2019.

[21] X. He, Y. Hua, T. Song, Z. Zhang, Z. Xue, R. Ma, N. Robertson, and H. Guan, "Unsupervised video summarization with attentive conditional generative adversarial networks," in *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, (New York, NY, USA), p. 2296–2304, Association for Computing Machinery, 2019.

[22] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pp. 505–520, Springer, 2014.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.