Extractive Video Summarizer with Memory Augmented Neural Networks

Litong Feng
SenseTime Research
fenglitong@sensetime.com

Ziyin Li
SenseTime Research
liziyin@sensetime.com

Zhanghui Kuang
SenseTime Research
kuangzhanghui@sensetime.com

Wei Zhang*
SenseTime Research
wayne.zhang@sensetime.com

ABSTRACT

Online videos have been growing explosively in recent years. How to help human users efficiently browse videos becomes more and more important. Video summarization can automatically shorten a video through extracting key-shots from the raw video, which is helpful for digesting video data. State-of-the-art supervised video summarization algorithms directly learn from manually-created summaries to mimic the key-frame/key-shot selection criterion of humans. Humans usually create a summary after viewing and understanding the whole video, and the *global* attention mechanism capturing information from all video frames plays a key role in the summarization process. However, previous supervised approaches ignored the temporal relations or simply modeled local inter-dependency across frames. Motivated by this observation, we proposed a memory augmented extractive video summarizer, which utilizes an external memory to record visual information of the whole video with high capacity. With the external memory, the video summarizer simply predicts the importance score of a video shot based on the global understanding of the video frames. The proposed method outperforms previous state-of-the-art algorithms on the public SumMe and TVSum datasets. More importantly, we demonstrate that the global attention modeling has two advantages: good transferring ability across datasets and high robustness to noisy videos.

CCS CONCEPTS

• **Information systems** → *Multimedia information systems*;

KEYWORDS

Video summarization; Memory networks; Global attention

ACM Reference Format:
Litong Feng, Ziyin Li, Zhanghui Kuang, and Wei Zhang. 2018. Extractive Video Summarizer with Memory Augmented Neural Networks. In *2018 ACM Multimedia Conference (MM '18), October22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3240508.3240651

*Corresponding author is Wei Zhang.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240651

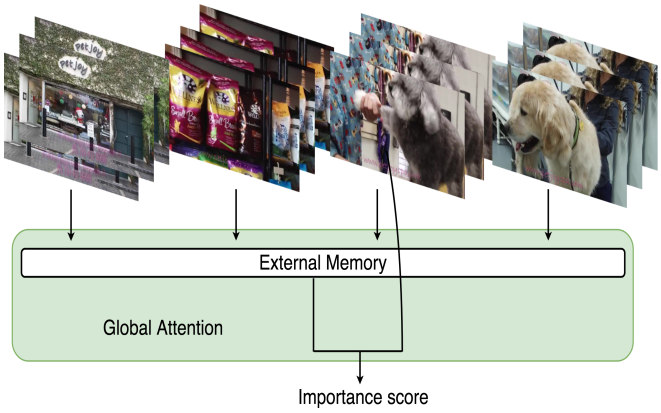

Figure 1: Global attention mechanism for video summarization.

1 INTRODUCTION

The amount of videos on the Internet is growing at a rapid pace, e.g., 300 hours of videos are uploaded to YouTube every minute in 2018 [39]. It brings the need of automatic video summarization to help human users browse videos efficiently. Video summarization aims at shortening a video through extracting key-shots from all shots in the original video.

A good video summary should not only keep a compact form but also cover the major content of the original video. Therefore, it is hard to decide whether to include a video shot in summary or not without the understanding of the whole video. A human video summary editor usually watches the original video first, understands the essential points or the story line, and creates the summary by deciding to keep or discard each shot in the video. During the process, the editor keeps the holistic understanding in his/her mind.

To mimic the process of video summary creation of humans, we utilizes a global attention mechanism (see Figure 1) to generate the importance score of a video shot conditioned on the whole input video. As the number of shots in a video varies from tens to thousands, learning a global attention from the full video requires a large memory. Inspired by memory networks for tasks such as question answering (QA) [12, 32], an external memory is designed to

record visual information of all shots, and training both the memory representation and importance score predictor can be integrated into our end-to-end memory augmented neural networks model.

Supervised video summarization [16, 43], which uses human-annotated summaries as guidance, achieved much better performance than unsupervised video summarization [8, 23, 25]. Previous supervised methods focused on either predicting frame-level or shot-level importance by learning the ground truth [26, 6, 7, 16], or optimizing a global objective of subset selection problem [5]. Most of the importance prediction methods considered neither dependencies between video shots nor attention from the whole video. Although long short-term memory (LSTM) has already been introduced to model the sequential dependencies across consecutive video shots [43, 14], the memorization capability of memory cells in LSTM is not as strong as external memory and the *global* attention has not been explored.

A memory network based approach was proposed for query-focused video summarization previously [27]. Different from the classical video summarization problem, query-focused video summarization introduced user preferences in the form of text queries about the video. In [27], the shot-level importance was measured by the relevance between the query and the shot. The memory networks were applied to each single shot, to obtain the importance score of the shot conditioned on the text query. The memory was used to store visual information of frames in a single shot and model the context within the shot. In comparison, our summarizer measures the shot-level importance based on the holistic understanding of the video. Our memory augmented neural networks are applied to the whole video, and the global attention is implemented by taking all shot features as input.

Our contributions in this paper are highlighted as follows: We propose a memory augmented video summarizer (MAVS), with the global attention mechanism and an external memory. The external memory with flexibility and high capacity facilitates global understanding of a video, and therefore, MAVS outperforms the state-of-the-art methods on the public SumMe [6] and TVSum [31] datasets in experiments. More importantly, the global attention modeling has two advantages: good transferring ability across datasets and high robustness to noisy videos, as demonstrated in experiments.

2 RELATED WORK

2.1 Video Summarization

Unsupervised Video Summarization. Unsupervised approaches focus on hand-crafted objectives for selecting keyframes/keyshots without the guidance of human-created ground truths. Different objectives or assumptions derive different algorithms. In the perspective of removing redundancy, clustering is a classic video summarization solution, which can capture sparse keyshots [8, 23, 25]. Although the whole video was taken as input, they aimed at obtaining super segments of the raw video other than understanding it. In the perspective of side information, Chu et al. assumed that highlight-frames had co-occurrence patterns among human-edited videos with the same topic. A video could be summarized through selecting shots that co-occurred most frequently across videos with the same topic name, retrieved from the search engine [2]. Ji et al. supposed that keyframes were highly salient. All the collected salient frames were further filtered using on-line clustering to remove redundant information [13]. Assuming that the story plot was played by roles in a movie, Tsai et al. utilized social networks to evaluate the social power of each role-communication, and the centrality values in social networks were utilized for ranking keyshots [37]. In the perspective of aesthetic, Song et al. selected keyframes according to a hand-crafted attractiveness score after removing redundant frames using clustering [30]. Zhang et al. assumed that motions of key objects must be captured by the keyframes. Trajectories of moving object instances were extracted and summarized through on-line auto-encoding, which can select spare and salient object motion-clips [44]. Recently, generative adversarial network (GAN) has been applied to video summarization, the summarizer is an auto-encoder for selecting sparse keyframes to reconstruct the raw video, where the discriminator distinguished between the original video and its reconstruction from the summary [20]. Reinforcement learning (RL) has also been utilized for video summarization. Zhou et al. treated video summarization as a sequential decision-making process, and optimized the action mechanism with a diversity-representativeness reward [46]. Although various unsupervised approaches have been proposed, it is difficult to fit the complicated video summarization process of humans with hand-crafted objectives.

Supervised Video Summarization. The quality of video summaries should be subjectively evaluated by human. Hence learning from human-created summary ground truths is a natural solution. Potapov et al. utilized support vector machine (SVM) to predict the importance score of every video shot, and video shots with the highest scores were assembled as a sequence to generate a summary [26]. Gygli et al. proposed a supervised approach to learn characteristics of a summary [7]. Gong et al. applied sequential e determinantal point process (DPP) to select diverse subsets, and feature embedding was learned with the guidance of manual summary labels [5]. All of the above early works used hand-crafted features for videos shot representation [19]. With the aid of deep convolutional neural network (CNN) features, the performance of video summarization can be improved a lot. In [16], a retrieval-based approach was utilized for video summarization, the semantic importance of each video shot was inferred based on the matched video shots in the training set, and Viterbi method was applied to keep the temporal coherence of video shots in the summary. Zhang et al. assumed that similar videos share similar summary structures. Thus, summary structures of training videos were represented as DPP kernels, and nonparametrically transferred to inference videos via pairwise matching [42]. Exploiting the temporal dependency among video frames or shots is very important for the task of video summarization. LSTM was combined with DPP to obtain both representative and compact summaries [43]. In order to explore a long temporal dependency between shots, a hierarchical LSTM was proposed for capturing multi-scale temporal dependencies [45]. On public datasets [6, 31], the state-of-art performance is achieved by these supervised methods. The core of recent supervised video summarization is how to design neural networks which can comprehend the video structure and the shot content simultaneously.

2.2 Memory Networks

Memory networks are learning models deliberated for QA tasks [1, 24, 36]. The long-term external memory acts as a knowledge base which can be read and written to. Reasoning can be incorporated with global attention over memory. A number of seminal works investigated ways to capture the long-term temporal attention within sequences using recurrent neural network (RNN) or LSTM-based models [9, 18, 22]. The LSTM-based models address sequential attention using local memory cells which lock the past state. This kind of temporal attention is latent and not global. Compared with LSTM, memory networks directly look-up to the knowledge base (memory) rather than rely on sequential state encoded by LSTM. Hence memory networks perform better at solving abstract reasoning problems which need multiple supporting knowledge without explicit temporal structures. Video summarization is a abstract computer vision task, without explicit vision patterns or semantic rules. Moreover, the summarization quality is evaluated by humans. Therefore, our strategy is to investigate an extractive video summarizer augmented with memory networks to mimic the human annotator, instead of making any explicit or implicit assumption of video summarization.

3 APPROACH

Video summarization is typically formulated as a subset selection problem [6]: First, the raw video is partitioned into many video shots. Video shot acts as the basic semantic unit in a video. Shot transitions edited by human correspond to abrupt or gradual semantic breaks. Frames within the same shot have high temporal coherence describing a sub-event[40]. Then, a learning-based or heuristic model predicts importance scores of all video shots, which depict their relevance to the final summary. Then, an optimization model is applied to select a summary from the shots based on their importance scores. For simplicity, we follow this standard process, and use a 0/1-knapsack optimization. More complicated optimization model such as sequential DPP may also be combined with our method. In this paper, we focus on designing the neural network model for predicting shot-level importance scores.

3.1 Shot Feature Representation

The input of the neural network model is shot features. Zhang et al. showed that deep convolutional features consistently improved performance over the hand-crafted features in video summarization [43]. As consecutive frames in a shot share much redundant semantic information, in this paper we simply average deep features of all frames within a shot to represent the video shot. Theoretically we can feed frame features into the neural networks, but shot-level feature representation is much more memory efficient, and makes it feasible to record the complete video information with an external memory.

We suppose that our model can be further improved in two directions: (1) Integrating better video feature representations. Average pooled deep features mainly carry appearance information such as scenes and objects depicted in the video, but miss motion information [28]. Advanced video feature representations have been extensively in video classification [10, 33, 38]. In this paper, we simply use the average pooled deep features for shot feature representation to demonstrate the capability of the prediction model. (2) End-to-end training with CNN. Many computer vision tasks such as video action recognition [28, 33, 17] benefit from end-to-end training with CNN. As there is no large-scale video summarization dataset, we have to rely on CNN models pretrained on large-scale image/video datasets such as ImageNet [3]. Deep convolutional feature extracted from frames using a pretrained image classification model is proved to be an effective feature solution.

3.2 Memory Augmented Video Summarizer

As discussed above, a video is segmented into video shots by shot boundary detection. The feature representing the i-th video shot is denoted as x_i, which is a vector with a dimension of v. The core procedure of video summarization is to infer an importance score for each video shot given $\{x_k\}$.

We illustrate the architecture of our proposed MAVS in Figure 2. To facilitate holistic understanding of the video, all $\{x_k\}$ are to be written into the external memory, consisting of an input memory and an output memory. The input memory is designed for providing supporting knowledge extracted from the whole video. The inner-product operation between the input memory $\{a_k\}$ and the shot feature x_i is to locate supporting facts relevant with the current shot. Each supporting fact is assigned with a relevance weight expressed as p_i^k. These relevance weights are further operated on the output memory for generating the global attention for the current shot. This global attention adjusts the importance score prediction of the current shot with holistic understanding of the raw video. In this way, a global attention mechanism is implemented and can be trained in an end-to-end fashion with human annotations. We describe the model mathematically as follows.

First, features $\{x_k\}$ are converted into the input memory vectors $\{a_k\}$ using an embedding matrix A (of size $d \times v$). When predicting an importance score for each video shot, the shot feature x_i is embedded to an internal state u_i with the same dimension of d as the input memory vectors $\{a_k\}$ using another embedding matrix U. In the embedding space, the match between u_i and $\{a_k\}$ are computed by inner-product followed by a softmax function as

$$p_i^k = Softmax(u_i^T a_k) \tag{1}$$

As defined in Equation (1), p_i^k is a probability vector over the input memory $\{a_k\}$ given u_i.

For producing the memory output o_i, an embedding matrix B converts features $\{x_k\}$ to the output memory vectors $\{b_k\}$. The memory output o_i is a sum over $\{b_k\}$ weighted by the probability vector p_i^k as

$$o_i = \sum_i p_i^k b_k \tag{2}$$

The final importance score s_i for each video shot will be regressed from the element-wise multiplication of internal state u_i and the memory output o_i using a fully connected layer D.

$$u_i' = u_i \odot o_i \tag{3}$$

$$s_i = W_D \cdot u_i' + b_D \tag{4}$$

The Equations (1)-(3) describe one computational step in the memory network framework. Multiple computational steps, termed

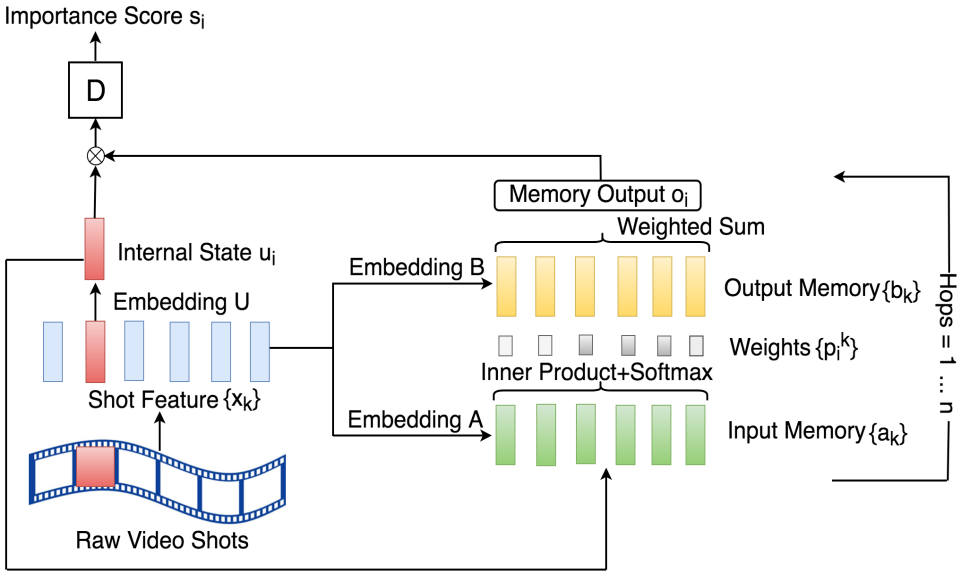

Figure 2: The architecture of the proposed memory augmented video summarizer.

as "hops", can be applied for pursuing a better performance on video summarization. With multiple hops, the adjacent weight tying is adopted[32], which means the input memory embedding matrix A in current hop is identical to the output memory embedding matrix B in the previous hop. The share of embedding parameters is helpful for small-size datasets.

Feature embedding transfers deep features from image classification to video summarization. As shown in Figure 2, all the operations in MAVS are differential. Hence, all the embedding functions can be optimized through back propagation.

3.3 Baselines

Multi-layer perceptron (MLP). In some work, the embedded video feature has been directly used to predict importance scores[6, 26]. MLP is considered as a baseline method in the experimental comparison.

Long short-term memory. LSTM is a special kind of RNN designed for learning long-term dependencies[9]. For each basic unit of LSTM, the hidden state h_i is output given the input x_i. The internal memory is recorded in cell state C_i, to which LSTM is able to add or remove information by gates. There are three different gates controlling the memory flow: the forget gate f deciding what to be removed from the cell state, the input gate i controlling what to be stored in the cell state, and the output gate o performs a cell state based attention on the output hidden state h_i. The detailed operations are given in Equation (5).

$$\begin{aligned} f_i &= \sigma(W_f \cdot [h_{i-1}, x_i] + b_f) \\ i_i &= \sigma(W_i \cdot [h_{i-1}, x_i] + b_i) \\ C_i &= f_i \cdot C_{i-1} + i_i \cdot tanh(W_C \cdot [h_{i-1}, x_i] + b_C) \\ h_i &= \sigma(W_o \cdot [h_{i-1}, x_i] + b_o) \cdot tanh(C_i) \end{aligned} \tag{5}$$

where, σ is the sigmoid function, $tanh$ is the hyperbolic tangent function, f_i, i_i are the outputs of the forget gate f and the input gate i given input x_i, C_i is the cell state updated at the step of x_i, and h_i is the output hidden state corresponding to x_i controlled by the output gate o.

The memory recorded in the cell sate is latently updated along with sequential data. This memory ability makes LSTM perform well on tasks dealing with sequential data, e.g. language translation and action recognition[4],[34]. LSTM is also considered as a baseline method in the experimental comparison.

4 EXPERIMENTS

We conducted our experiments on two popular video summarization datasets: SumMe [6] and TVSum [31]. Three kinds of experiments are performed, consisting of intra-dataset video summarization, inter-dataset video summarization, and noisy video summarization.

4.1 Datasets

4.1.1 SumMe [6]. The SumMe dataset consists of 25 user videos covering holidays, events, and sports. There is no category information in SumMe. The video length varies from 1 minute to 6 minutes. All the 25 videos belong to different categories. Frame-level importance sores for video summarization labelled by human are provided. Each video is labelled by multiple annotators.

4.1.2 TVSum [31]. The TVSum dataset contains 50 videos downloaded from YouTube. The video length varies from 2 minutes to 10 minutes. All the 50 videos belong to 10 categories (5 per category) defined in TRECVid Multimedia Event Detection (MED) task [29]. The videos are searched on Youtube using the category name as a query. Human created frame-level importance sores for video summarization are given. Each video has multiple manual labels.

4.2 Experimental Settings

4.2.1 Shot Segmentation. The video shot boundary is detected using an adaptive thresholding based shot boundary detection algorithm[41], except that the hand-crafted feature is replaced with a CNN feature for a better performance. Multi-temporal-scale operation is performed to solve cut and gradual transitions simultaneously. The cosine distance between normalized CNN features is used to measure frame similarity. And the CNN feature is extracted using SqueezeNet for a fast speed [11].

4.2.2 Shot Features. For a fair comparison with other recent work [42, 43], the feature describing each video frame utilizes the deep CNN feature extracted at the penultimate layer (pool5) of GoogLeNet [35], with a dimension of 1024. And this deep feature is L2-normalized per frame[15]. In all experiments, we use the shot-level feature, which is the average of frame-level features within the same video shot.

4.2.3 Evaluation Metrics. Following the protocol in previous work[6],[31], the length of the generated video summary S is no longer than 15% of the raw video. Precision P and Recall R are calculated against the summary ground truth T, and their harmonic mean F-score is treated as the final evaluation metric, as depicted in Equation (6).

$$P = \frac{S \cap T}{S}$$
$$R = \frac{S \cap T}{T} \tag{6}$$
$$F = \frac{2P \times R}{P + R}$$

Both the predicted video summary and the ground truth summary are collected based on importance scores by solving the knapsack problem. And 15% is set as the knapsack capacity. For processing multiple human-annotated summaries of a video, the metric calculation is followed as in [42].

4.2.4 Hyper-parameters in Model Training. A linear L2 regression loss is used for fitting the importance score. All parameters learning is implemented using back-propagation. The initial learning rate is set as 0.01. Adam is selected as the optimization algorithm. The visual feature extraction is implemented using a fixed CNN model pre-trained on the ImageNet dataset. And the number of trainable parameters of MAVS are small. Hence no drop-out regularization is needed. The batch size is 300. The epochs for training MAVS and MLP are 20. The epochs for training LSTM are 40. The embedding size for MAVS is 512, and hops of MAVS vary from 1 to 6. The memory size of MAVS is not limited. It is set to cover the longest video in a dataset. The typical memory size used in this paper is 300. If a video does not have enough shots to fill the memory, the left empty memory will be filled with zero-vectors. The number of hidden layers of MLP varies from 1 to 4. Each hidden layer has a dimension of 512. The dimension of hidden state in LSTM is also set as 512. The number of time steps of LSTM is set as 4. Stacked LSTM is used, with layers varying from 1 to 3. Due to the limited size of SumMe and TVSum, the convergence of model optimization does not fall to a very stable point. Hence a validation set is used for selecting a best model from the last 5 models saved in the last 5 epochs. Train, validation, and test sets are partitioned with a ratio of 3:1:1. The validation set can be only used in the intra-dataset experiment. For the inter-dataset experiment, the validation set is useless due to different data distributions between SumMe and TVSum. The model training and inference were implemented based on TensorFlow, with a GTX TITAN Xp GPU.

4.2.5 Intra-dataset Video Summarization. The model training and model testing utilize data from the same dataset. This dataset setting is named as intra-dataset. The intra-dataset experiment was performed on both SumMe and TVSum. A dataset will be partitioned into 5 folds. Three folds are for the train set, and one fold for the validation set and the test set, respectively. Then 5-fold cross validation will be performed 5 times. The average F-score of all cross validations is reported as the final result. The comparison methods include LSTM [43], summary-transfer [42], temporal-tessellation [16], RL-based [46], and GAN-based [20].

4.2.6 Inter-dataset Video Summarization. Train data and test data come from different datasets. This dataset setting is named as inter-dataset. The model will be trained on one dataset and be tested on the other dataset. This setting is for testing the generalization ability of models across different datasets. The cross-dataset transferring ability of MAVS will be compared with MLP and LSTM.

4.2.7 Noisy Video Summarization. We will challenge the model robustness with a further step. For the inter-dataset experiments, videos in SumMe and TVSum are disturbed by other irrelevant video content. The irrelevant videos are movie trailers downloaded from YouTube, covering differ genres, including romance, action, sports, comedy, science-fiction. The ambition in trailer-making is to impart an intriguing story making audiences emotionally involved [21]. In addition to visually attractive shots, other appealing information is also collected for making a trail, e.g. funny conversations, cast-run, production logos, descriptive texts, etc. Compared with user videos in SumMe and TVSum, information in trailers are more disordered, thus we chose movie trailers to disturb raw videos in SumMe and TVSum. Each raw video was inserted with additional 25% irrelevant noisy shots. And noisy video shots were randomly placed into the raw video shots. The original sequence of raw video shots was not changed. In noisy data experiments, the models are still trained on original videos in SumMe and TVSum, but they will be tested on the generated noisy data.

4.3 Results and Analyses

In this section, we will show the comparison results with different experimental settings. Then, a typical summary case will be visualized. In addition, the visualization of global attentions learned by MAVS will be given.

4.3.1 Intra-dataset Video Summarization. The performance comparison of intra-dataset video summarization is shown in Table 1. On both SumMe and TVSum, MAVS achieved the best performance. It is observed that the performance of MAVS can be improved with the increase of hops. Specially, SumMe gained more with the hops increase compared with TVSum. There is no category information in SumMe, videos share less highlight patterns compared with the category-specific TVSum. Hence, memory networks seem to gain

more in dealing with category-unspecific data, which is more complex. More hops provide more opportunities to write and read the memory. When predicting an importance score for each video shot, the internal state can be operated with external memories several times to learn different global attentions. Hence, more hops gain improved performance. However, the gain due to more hops got a saturation at hops of 4. MAVS uses the average frame feature of a shot to represent the shot content. Hence MAVS relies on shot segmentation. TVSum has a higher annotation-quality than SumMe, with shots segmented for annotation. TVSum is more suitable for representing video semantics using shots. Hence, MAVS performed better on TVSum than SumMe in comparison with other methods.

Methods	hops	SumMe	TVSum
DPP-LSTM (Canonical)[43]	-	38.6	54.7
Summary-transfer[42]	-	40.9	-
GAN-based (SUM-GAN$_{sup}$)[20]	-	41.7	56.3
RL-based (DR-DSN$_{sup}$)[46]	-	42.1	58.1
Temporal-tessellation (Unsupervised)[16]	-	41.4	64.1
MAVS	1	39.8	67.0
MAVS	2	42.5	67.2
MAVS	3	42.6	67.3
MAVS	4	43.1	67.5
MAVS	5	42.3	67.3
MAVS	6	40.3	66.8

Table 1: Performance comparison of intra-dataset video summarization on SumMe and TVSum.

Methods	hops	SumMe2TVSum	TVSum2SumMe
MLP1	-	64.5	36.5
MLP2	-	62.1	35.5
MLP3	-	61.7	36.4
MLP4	-	61.4	35.8
LSTM1	-	62.8	36.6
LSTM2	-	61.5	36.9
LSTM3	-	60.1	34.6
MAVS	1	63.2	37.3
MAVS	2	65.5	38.8
MAVS	3	66.2	39.8
MAVS	4	66.4	41.7
MAVS	5	66.3	41.2
MAVS	6	65.8	40.2

Table 2: Performance comparison of inter-dataset video summarization on SumMe [6] and TVSum [31].

4.3.2 Inter-dataset Video Summarization. The performance comparison of inter-dataset video summarization is shown in Table 2. SumMe2TVSum means that the model is trained on SumMe and tested on TVSum. And TVSum2SumMe means the opposite setting. MAVS performed better than MLP and LSTM. The global attention in MAVS can improve the transfer abilities of models across datasets. For the inter-dataset experiments, the hops increase of MAVS also gained better performance. The increase of layers for MLP and LSTM got no obvious gain.

4.3.3 Noisy Video Summarization. The noisy video data much affected the summarization performance, as shown in Table 3. Though F-scores of all methods in comparison decreased, MAVS still obtained a better performance than MLP and LSTM. LSTM had a better result on TVSum2SumMe compared with MLP. While MLP over-performed LSTM on SumMe2TVSum.

Methods	hops	SumMe2TVSum	TVSum2SumMe
MLP1	-	45.1	18.8
MLP2	-	47.6	19.7
MLP3	-	47.9	18.4
MLP4	-	46.1	18.8
LSTM1	-	40.7	24.0
LSTM2	-	41.7	24.6
LSTM3	-	41.1	24.2
MAVS	1	50.8	19.1
MAVS	2	57.2	20.7
MAVS	3	57.9	22.6
MAVS	4	59.4	25.9
MAVS	5	58.3	25.3
MAVS	6	58.5	25.4

Table 3: Performance comparison of noisy video summarization with inter-dataset experimental setting.

4.3.4 Qualitative Result. A typical video summarization result is visualized in Figure 3. This is the result of video "-esJrBWj2d8" in TVSum for the inter-dataset experiment. Blue bars correspond to the ground truth importance scores. And yellow bars correspond to the selected summaries by MAVS(4-hops), LSTM1, and MLP2 methods, respectively. Frames showed in Figure 3 are key-frames representing their original video shots. Key-frames in green rectangles are true positive cases. Key-frames in red rectangles are false positive cases. It can be observed that all true positive cases are relevant with feeding the cat or the cat's movements. And the cat or the feeding action is not obvious in false positive cases. In this case of Figure 3, MAVS is able to select two more positive video shots compared with both MLP and LSTM.

We have the motivation to make MAVS able to learn global attentions from the whole video. What are the global attentions like? It is hard to analyze attention by directly visualizing features. We selected to visualize the p_i^k in Equation (1) over memories. The softmax result p_i^k can reflect what parts of the raw video on which every video shots put more weights to generate the memory output. And memory outputs correspond to global attentions in different hops. With the SumMe2TVSum setting in inter-dataset experiments, the averaged p_i^k of all video shots in a video ("cjibtmSLxQ4", F-score 64.9) is given in Figure 4. A 4-hops MAVS is applied here. It can be observed that different hops actually focus on different parts of the video. And this sofrmax attention in MAVS is sparse, like finding the major points in the raw video.

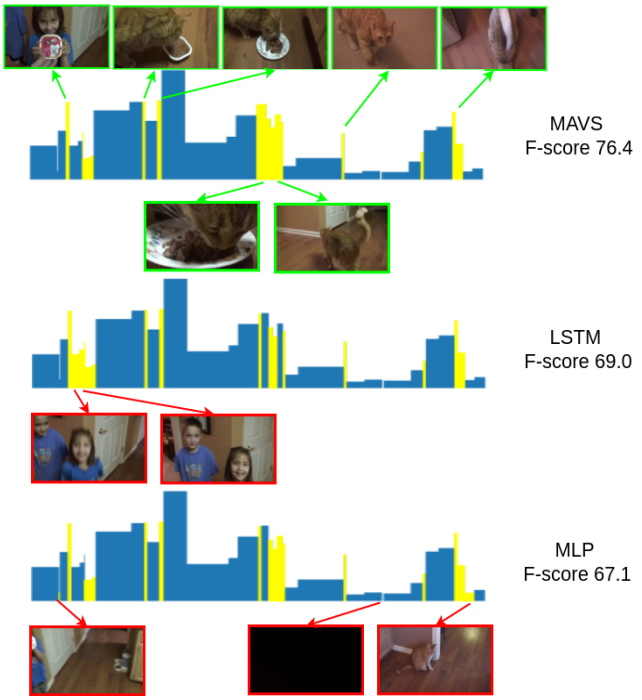

Figure 3: Visualization of a typical case in the inter-dataset experiment. (Green rectangle - True positive, Red rectangle - False positive)

Figure 4: Visualization of typical softmax attentions in MAVS.

5 CONCLUSION

Video summarization is a very abstract computer vision task. The objective is hard to be defined to match the human's subjective understanding in the video summary creation. Hence, mimicking the human-labelled summaries is a promising approach to solve this problem. Summary creation should follow the holistic understanding of the raw video. We proposed a memory augmented video summarizer (MAVS) to predict the importance score of each video shot for the final video summary. The memory networks consist of an external memory loading the whole video information, which can effectively learn the global attention from the whole video to assist understanding the importance of each video shot. On the public SumMe and TVSum datasets, the proposed MAVS achieved leading performance. For challenging the robustness of the MAVS model, inter-dataset experiments showed its good cross-dataset transferring ability. On the disturbed SumMe and TVSum, MAVS still had a better transfer performance compared with LSTM and MLP.

We would like to explore in two directions in the future: (1) we will investigate better shot feature representation, either human-designed features or CNN features by end-to-end training on a large video summarization dataset. (2) We only solved video summarization using memory networks with visual information. Other modalities, such as sounds and subtitles, also play a key role in summarization. We will investigate to use memory networks with multiple modalities to further improve the automatic video summarization.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

[2] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. 2015. Video co-summarization: video summarization by visual co-occurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3584–3592.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: a large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2625–2634.

[5] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. 2014. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, 2069–2077.

[6] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *European conference on computer vision*. Springer, 505–520.

[7] Michael Gygli, Helmut Grabner, and Luc Van Gool. 2015. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 3090–3098.

[8] Youssef Hadi, Fedwa Essannouni, and Rachid Oulad Haj Thami. 2006. Video summarization by k-medoid clustering. In *Proceedings of the 2006 ACM symposium on Applied computing*. ACM, 1400–1401.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9, 8, 1735–1780.

[10] Po-Yao Huang, Ye Yuan, Zhenzhong Lan, Lu Jiang, and Alexander G Hauptmann. 2017. Video representation learning and latent concept mining for large-scale multi-label video classification. *arXiv preprint arXiv:1707.01408*.

[11] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. Squeezenet: alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.

[12] A. Bordes J. Weston S. Chopra. [n. d.] Memory networks. In *2015 ICLR*.

[13] Qing-Ge Ji, Zhi-Dang Fang, Zhen-Hua Xie, and Zhe-Ming Lu. 2013. Video abstraction based on the visual attention model and online clustering. *Signal Processing: Image Communication*, 28, 3, 241–253.

[14] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. 2017. Video summarization with attention-based encoder-decoder networks. *arXiv preprint arXiv:1708.09545*.

[15] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision*. Springer, 685–701.

[16] Dotan Kaufman, Gil Levi, Tal Hassner, and Lior Wolf. 2017. Temporal tessellation: a unified approach for video analysis. In *The IEEE International Conference on Computer Vision*. Vol. 8.

[17] Will Kay et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

[18] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. 2014. A clockwork rnn. *arXiv preprint arXiv:1402.3511*.

[19] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 2, 91–110.

[20] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. 2017. Unsupervised video summarization with adversarial LSTM networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[21] Robert Marich. 2013. *Marketing to moviegoers: a handbook of strategies and tactics*. SIU Press.

[22] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, and Marc'Aurelio Ranzato. 2014. Learning longer memory in recurrent neural networks. *arXiv preprint arXiv:1412.7753*.

[23] Padmavathi Mundur, Yong Rao, and Yelena Yesha. 2006. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6, 2, 219–232.

[24] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. 2017. A read-write memory network for movie story understanding. *arXiv preprint arXiv:1709.09345*.

[25] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. 2005. Video summarization and scene detection by graph modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 15, 2, 296–305.

[26] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. 2014. Category-specific video summarization. In *European conference on computer vision*. Springer, 540–555.

[27] Aidean Sharghi, Jacob S Laurel, and Boqing Gong. 2017. Query-focused video summarization: dataset, evaluation, and a memory network based approach. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2127–2136.

[28] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576.

[29] Alan F Smeaton, Paul Over, and Wessel Kraaij. 2006. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*. ACM, 321–330.

[30] Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. 2016. To click or not to click: automatic selection of beautiful thumbnails from videos. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 659–668.

[31] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5179–5187.

[32] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448.

[33] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. 2018. Optical flow guided feature: a fast and robust motion representation for video action recognition.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

[36] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: understanding stories in movies through question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4631–4640.

[37] Chia-Ming Tsai, Li-Wei Kang, Chia-Wen Lin, and Weisi Lin. 2013. Scene-based movie summarization via role-community networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 23, 11, 1927–1940.

[38] Jue Wang, Anoop Cherian, Fatih Porikli, and Stephen Gould. 2018. Video representation learning using discriminative pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1149–1158.

[39] 2018. Youtube statistics. https://fortunelords.com/youtube-statistics/. (2018).

[40] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang. 2007. A formal study of shot boundary detection. *IEEE transactions on circuits and systems for video technology*, 17, 2, 168–186.

[41] Yusseri Yusoff, William J Christmas, and Josef Kittler. 2000. Video shot cut detection using adaptive thresholding. In *BMVC*, 1–10.

[42] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Summary transfer: exemplar-based subset selection for video summarization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 1059–1067.

[43] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *European Conference on Computer Vision*. Springer, 766–782.

[44] Yujia Zhang, Xiaodan Liang, Dingwen Zhang, Min Tan, and Eric P Xing. 2018. Unsupervised object-level video summarization with online motion auto-encoder. *arXiv preprint arXiv:1801.00543*.

[45] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2017. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 863–871.

[46] Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *The Thirty-Second AAAI Conference on Artificial Intelligence*.