Improving Sequential Determinantal Point Processes for Supervised Video Summarization

Aidean Sharghi[1](✉), Ali Borji[1](✉), Chengtao Li[2](✉), Tianbao Yang[3](✉), and Boqing Gong[4](✉)

[1] Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA
aidean.sharghi@gmail.com, aliborji@gmail.com

[2] Massachusetts Institute of Technology, Cambridge, MA, USA
ctli@mit.edu

[3] University of Iowa, Iowa City, IA, USA
tianbao-yang@uiowa.edu

[4] Tencent AI Lab, Seattle, WA, USA
boqinggo@outlook.com

Abstract. It is now much easier than ever before to produce videos. While the ubiquitous video data is a great source for information discovery and extraction, the computational challenges are unparalleled. Automatically summarizing the videos has become a substantial need for browsing, searching, and indexing visual content. This paper is in the vein of supervised video summarization using sequential determinantal point processes (SeqDPPs), which models diversity by a probabilistic distribution. We improve this model in two folds. In terms of learning, we propose a large-margin algorithm to address the exposure bias problem in SeqDPP. In terms of modeling, we design a new probabilistic distribution such that, when it is integrated into SeqDPP, the resulting model accepts user input about the expected length of the summary. Moreover, we also significantly extend a popular video summarization dataset by (1) more egocentric videos, (2) dense user annotations, and (3) a refined evaluation scheme. We conduct extensive experiments on this dataset (about 60 h of videos in total) and compare our approach to several competitive baselines.

1 Introduction

It is now much easier than ever before to produce videos due to ubiquitous acquisition capabilities. The videos captured by UAVs and drones, from ground surveillance, and by body-worn cameras can easily reach the scale of gigabytes per day. In 2017, it was estimated that there were at least 2.32 billion active

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01219-9_32) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11207, pp. 533–550, 2018.
https://doi.org/10.1007/978-3-030-01219-9_32

camera phones in the world [24]. In 2015, 2.4 million GoPro body cameras were sold worldwide [13]. While the big video data is a great source for information discovery and extraction, the computational challenges are unparalleled. Automatically summarizing the videos has become a substantial need for browsing, searching, and indexing visual content.

Under the *extractive* video summarization framework, a summary is composed of important shots of the underlying video. This notion of importance, however, varies drastically from work to work in the literature. Wolf defines the importance as a function of motion cues [41]. Zhao and Xing formulate it by reconstruction errors [47]. Gygli et al. learn a mixture of *interestingness*, *representativeness*, and *uniformity* measures to find what is important [12]. These differences highlight the complexity of video summarization. The criteria for summarizing vastly depend on the content, styles, lengths, etc. of the video and, perhaps more importantly, users' preferences. For instance, to summarize a surveillance video, a running action might flag an important event whereas in a football match it is a normal action observed throughout the video.

To overcome those challenges, there are two broad categories of approaches in the literature. One is to constrain the problem domain to a homogeneous set of videos which share about the same characteristics (e.g., length and style) so that experts can engineer some domain-specific criteria of good summaries [26,34]. The other is to design models that can learn the criteria automatically, often from human-annotated summaries in a supervised manner [9,30,31,46]. The latter is more appealing because a learner can be trained for different settings of choice, while the former is not so scalable.

This paper is also in the vein of supervised video summarization based on determinantal point processes (DPPs) [18]. Arising from quantum physics and random matrix theories, DPP is a powerful tool to balance importance and diversity, two axiomatic properties in extractive video summarization. Indeed, a good summary must be collectively diverse in the sense that it should not have redundancy of information. Moreover, a shot selected into the summary must add value to the quality of the summary; otherwise, it is not important in the context of the summary. Thanks to the versatility of DPP and one of its extensions called SeqDPP [9] for handling sequences, they have been employed in a rich line of recent works on video summarization [30,31].

This paper makes two-pronged contribution, improving these models from the perspectives of both model flexibility and learning strategy. In terms of learning, we propose a large-margin algorithm to address the SeqDPP's exposure bias problem explained below. In terms of modeling, we design a new probabilistic block such that, when it is integrated into SeqDPP, the resulting model accepts user input about the expected length of the summary.

We first explain the exposure bias problem with the existing SeqDPP works—it is actually a mismatch issue in many sequence to sequence (seq2seq) learning methods [1,29,35,36,39]. When the model is trained by maximizing the likelihood of user annotations, the model takes as input user annotated "oracle" summaries. At the test time, however, the model generates output by searching over the output space in a greedy fashion and its intermediate conditional distributions may receive input from the previous time step that deviates from the oracle. In other words, the model is exposed to different environments in the training and testing stages, respectively. This exposure bias also results in the loss-evaluation mismatch [27] between the training phase and the inference.

To tackle these issues, we adapt the *Large-Margin* algorithm originally derived for training LSTMs [40] to the SeqDPPs. The main idea is to alleviate the exposure bias by incorporating the techniques of the test time into the objective function used for training. Meanwhile, we add to the large-margin formulation a multiplicative reward term which is related to the evaluation metrics to mitigate the loss-evaluation mismatch.

In addition to the new large-margin learning algorithm, we also improve the SeqDPP model by a novel probabilistic distribution in order to allow users to control the lengths of system-generated video summaries. To this end, we propose a generalized DPP (GDPP) in which an arbitrary prior distribution can be imposed over the sizes of the subsets of video shots. As a result, both vanilla DPP and k-DPP [17] can be considered as special instances of GDPP. Moreover, we can conveniently substitute the (conditional) DPPs in SeqDPP by GDPP. When a user gives an expected length of the summary, we dynamically allocate it to different segments of the video and then choose the right number of video shots from a segment.

We conduct extensive experiments to verify the improved techniques for supervised video summarization. First of all, we significantly extend the UTE dataset [19], its annotations of video summaries, and per-shot concepts [31] by another eight egocentric videos [8]. Following the protocol described in [31], we collect three user summaries for each of the hours-long videos as well as concept annotations for each video shot. We evaluate the large-margin learning algorithm on not only the proposed sequential GDPP but also the existing SeqDPP models.

2 Related Work and Background

We briefly review the related work in this section. Besides, we also describe the major body of DPPs and SeqDPPs. Readers are referred to [18] and [9] for more details and properties of the two versatile probability models.

Supervised Video Summarization. In recent years, data-driven learning algorithms have prevailed in a variety of computer vision problems. This is mainly because they can learn complex relations from data, especially when the underlying relations are too subtle or complex to handcraft. Video summarization is an instance of such cases. The fact that different users prefer different summaries is a strong evidence to the complexity of the problem. To overcome the impediments, one solution is to learn how to make good summaries in a supervised manner. The degree of supervision, however, is different in the literature. In [4,15,16,42], weakly supervised web image and video priors help define visual importance. Captions associated with videos are used by [22,33] to infer semantic importance. Finally, many frameworks (e.g., [9,12,30,31,46]) learn a summarizer directly from user-annotated summaries.

Sequence-to-Sequence Learning. Sequence-to-sequence (Seq2seq) modeling has been successfully employed in a vast set of applications, especially in Natural Language Processing (NLP). By the use of Recurrent Neural Networks (RNNs), impressive modeling capabilities and results are achieved in various fields such as machine translation [1] and text generation applications (e.g., for image and video captioning [38,43]).

The Seq2seq models are conveniently trained as conditional language models, maximizing the probability of observing next ground truth word conditioned on the input and target words. This translates to using merely a word-level loss (usually a simple cross-entropy over the vocabulary).

While the training procedure described above has shown to be effective in various word-generation tasks, the learned models are not used as conditional models during inference at test time. Conventionally, a greedy approach is taken to generate the output sequence. Moreover, when evaluating, the complete output sequence is compared against the gold target sequence using a sequence-level evaluation metric such as ROUGE [21] and BLEU [25].

Determinantal Point Process (DPP). A discrete DPP [14,18] defines a distribution over all the subsets of a ground set measuring the negative correlation, or repulsion, of the elements in each subset. Given a ground set $\mathcal{Y} = \{1, \ldots, N\}$, one can define $K \in \mathbb{R}^{N \times N}$, a positive semi-definite kernel matrix that represents the per-element importance as well as the pairwise similarities between the N elements. A distribution over a random subset $Y \subseteq \mathcal{Y}$ is a DPP, if for every $y \subseteq \mathcal{Y}$ the following holds:

$$P(y \subseteq Y; K) = \det(K_y) \tag{1}$$

where K_y is the squared sub-kernel of K with rows and columns indexed by the elements in y, and $\det(.)$ is the determinant function. K is referred to as the marginal kernel since one can compute the probability of any subset y being included in $\mathcal{Y}$. It is the property of the determinant that promotes diversity: in order to have a high probability $P(i, j \in Y; K) = K_{ii}K_{jj} - K_{ij}^2$, the per-element importance terms K_{ii} and K_{jj} must be high and meanwhile the pairwise similarity terms K_{ij} must be low.

To directly specify the atomic probabilities for all the subsets of $\mathcal{Y}$, Borodin and Rains derived another form of DPPs through a positive semi-definite matrix $L = K(I - K)^{-1}$ [2], where I is an identity matrix. It samples a subset $y \subseteq \mathcal{Y}$ with probability

$$P_L(Y = y; L) = \frac{\det(L_y)}{\det(L + I)}, \tag{2}$$

where the denominator $\det(L + I)$ is a normalization constant.

Sequential DPP (seqDPP). Gong et al. proposed SeqDPP [10] to preserve partial orders of the elements in the ground set. Given a long sequence $\mathcal{V}$ of elements (e.g., video shots), we divide them into T disjoint yet consecutive partitions $\bigcup_{t=1}^{\mathsf{T}} \mathcal{V}_t = \mathcal{V}$. The elements within each partition are orderless to apply DPP and yet the orders between the partitions are observed in the following manner. At the t-th time step, SeqDPP selects a diverse subset of elements by a variable $X_t \subseteq \mathcal{V}_t$ from the corresponding partition and conditioned on the elements $x_{t-1} \subseteq \mathcal{V}_{t-1}$ selected from the previous partition. In particular, the distribution of the subset selection variable X_t is given by a conditional DPP,

$$P(X_t = x_t | X_{t-1} = x_{t-1}) := P_L(Y_t = x_t \cup x_{t-1} | x_{t-1} \subseteq Y_t; L^t) \tag{3}$$

$$= P_L(X_t = x_t; \Omega^t) = \frac{\det \Omega^t_{x_t}}{\det(\Omega^t + I)}, \tag{4}$$

where $P_L(Y_t; L^t)$ and $P_L(X_t; \Omega^t)$ are two L-ensemble DPPs with the ground sets $x_{t-1} \cup \mathcal{V}_t$ and $\mathcal{V}_t$, respectively—namely, the conditional DPP itself is a valid DPP over the "shrinked" ground set. The relationship between the two L-ensemble kernels L^t and Ω^t is given by [2],

$$\Omega^t = \left([(L^t + I_{\mathcal{V}_t})^{-1}]_{\mathcal{V}_t} \right)^{-1} - I, \tag{5}$$

where $I_{\mathcal{V}_t}$ is an identity matrix of the same size as L^t except that the diagonal entries corresponding to x_{t-1} are 0's, $[\cdot]_{\mathcal{V}_t}$ is the squared submatrix of $[\cdot]$ indexed by the elements in $\mathcal{V}_t$, and the number of rows/columns of the last identity matrix I equals the size of the t-th video segment $\mathcal{V}_t$.

3 A Large-Margin Algorithm for Learning SeqDPPs

We present the main large-margin learning algorithm in this section. We first review the mismatch between the training and inference of SeqDPPs [9] and then describe the large-margin algorithm in detail.

Training and Inference of SeqDPP. For the application of supervised video summarization, SeqDPP is trained by maximizing the likelihood (MLE) of user summaries. At the test time, however, an approximate online inference is employed:

$$\hat{x}_1 = \text{argmax}_{x \in \mathcal{V}_1} P(X_1 = \hat{x}), \quad \hat{x}_2 = \text{argmax}_{x \in \mathcal{V}_2} P(X_2 = \hat{x} | X_1 = \hat{x}_1), \quad \ldots \tag{6}$$

We note that, in the inference phase, a possible error at one time step (e.g., $\hat{x}_1$) propagates to the future but MLE always feeds the oracle summary to SeqDPP in the training stage (i.e., exposure bias [27]). Besides, the likelihood based objective function used in training does not necessarily correlate well with the evaluation metrics in the test stage (i.e., loss-evaluation mismatch [27]).

The issues above are common in seq2seq learning. It has been shown that improved results can be achieved if one tackles them explicitly [5,6,27,28,32]. Motivated by these findings, we propose a large-margin algorithm for SeqDPP to mitigate the exposure bias and loss-evaluation mismatch issues in existing SeqDPP works. Our algorithm is extended from [40], which studies the large-margin principle in training recurrent neural networks. However, we are not constrained by the beam search, do not need to change the probabilistic SeqDPP model to any non-probabilistic version, and also fit a test-time evaluation metric into the large-margin formulation.

We now design a loss function as the following,

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \delta(x^*_{1:t-1} \cup \hat{x}_t, x^*_{1:t}) M(x^*_t, \hat{x}_t, x^*_{t-1}; L), \tag{7}$$

which includes two components: (1) a sequence-level cost δ which allows us to scale the loss function depending on how erroneous the test-time inference is compared to the oracle summary, and (2) a margin-sensitive loss term M which penalizes the situation when the probability of an oracle sequence fails to exceed the probability of the model-inferred ones by a margin. Denote by $\hat{x}_t$ and $\hat{x}^*_t$ the subsets selected from the t-th partition $\mathcal{V}_t$ by SeqDPP and by an "oracle" user, respectively. Let $x^*_{1:t}$ represent the oracle summary *until* time step t. The sequence-level cost $\delta(x^*_{1:t-1} \cup \hat{x}_t, x^*_{1:t})$ can be any accuracy metric (e.g., 1-F-score) contrasting a system-generated summary with a user summary.

Assuming SeqDPP is able to choose the right subset x^*_{t-1} from partition $\mathcal{V}_{t-1}$, given the next partition $\mathcal{V}_t$, the margin-sensitive loss penalizes the situation that the model selects a different subset $\hat{x}_t$ from the oracle x^*_t,

$$\begin{aligned} M(x^*_t, \hat{x}_t, x^*_{t-1}; L) &:= [1 - \log P(X_t = x^*_t | x^*_{t-1}) + \log P(X_t = \hat{x}_t | x^*_{t-1})]_+ \\ &= [1 - \log\det(L_{x^*_t \cup x^*_{t-1}}) + \log\det(L_{\hat{x}_t \cup x^*_{t-1}})]_+ \end{aligned} \tag{8}$$

where $[\cdot]_+ = \max(\cdot, 0)$. When we use this loss term to train SeqDPP, we always assume that the correct subset $\hat{x}_{t-1} = x^*_{t-1}$ is chosen at the previous time step $t-1$. In other words, we penalize the model step by step instead of checking the whole sequence of subsets predicted by the model. This allows more effective training because it (1) enforces the model to choose the correct subset at every time step, and (2) enables us to set the gradient weights according to how erroneous a mistake is at a time step, rather than the whole sequence of all steps, in the eyes of the evaluation metric.

Compared to MLE, it is especially appealing that the large-margin formulation flexibly takes the evaluation metric into account. As a result, it does not require SeqDPP to predict exactly the same summaries as the oracles. Instead, when the predicted and oracle summaries are equivalent (not necessarily identical) according to the evaluation metric, the model parameters are not updated.

4 Disentangling Size and Content in SeqDPP

In this section, we propose a sequential model of generalized DPPs (Seq*G*DPP) that accepts an arbitrary distribution over the sizes of the subsets whose content follow DPP distributions. It allows users to provide priors or constraints over the total items to be selected. We first present the generalized DPP and then describe how to use it to devise the sequential model, Seq$G$DPP.

4.1 Generalized DPPs (GDPPs)

Kulesza and Taskar have made an intriguing observation about the vanilla DPP: it conflates the size and content of the variable Y for selecting subsets from the ground set $\mathcal{Y}$ [17]. To see this point more clearly, we can re-write a DPP as a mixture of elementary DPPs $P_E(Y)$ [18, Lemma 2.6],

$$P_L(Y; L) = \frac{1}{\det(L+I)} \sum_{J \subseteq \mathcal{Y}} P_E(Y; J) \prod_{n \in J} \lambda_n, \tag{9}$$

$$\propto \sum_{k=0}^{N} \sum_{J \subseteq \mathcal{Y}, |J|=k} P_E(Y; J) \prod_{n \in J} \lambda_n \tag{10}$$

where the first summation is over all the possible sizes of the subsets and the second is about the particular items of each subset. Eigen-decomposing the L-ensemble kernel to $L = \sum_{n=1}^{N} \lambda_n v_n v_n^T$, the marginal kernel of the elementary DPP $P_E(Y; J)$ is $K^J = \sum_{n \in J} v_n v_n^T$—it is interesting to note that, due to this form of the marginal kernel, the elementary DPPs do not have their counterpart L-ensembles. The elementary DPP $P_E(Y; J)$ always chooses $|J|$ items from the ground set $\mathcal{Y}$, namely, $P(|Y| = |J|) = 1$.

Equation (10) indicates that, to sample from the vanilla DPP, one may sample the size of a subset from a uniform distribution followed by drawing items/content for the subset. We propose to perturb this process and explicitly impose a distribution $\pi = \{\pi_k\}_{k=0}^{N}$ over the sizes of the subsets,

$$P_G(Y; L) \propto \sum_{k=0}^{N} \pi_k \sum_{J \subseteq \mathcal{Y}, |J|=k} P(Y; J) \prod_{n \in J} \lambda_n \tag{11}$$

As a result, the generalized DPP (GDPP) $P_G(Y; L)$ entails both DPP and k-DPP [17] as special cases (when π is uniform and when π is a Dirac delta distribution, respectively), offering a larger expressive spectrum. Another interesting result is that, for a truncated uniform distribution π over the sizes of the subsets, we arrive at a DPP which selects subsets with bounded cardinality, $P(Y \,|\, k_1 \le |Y| \le k_2; L)$. Such constraint arises from real applications like document summarization, image display, and sensor placement.

Normalization. The normalization constant for GDPP is $Z_G = \sum_{J \subseteq \mathcal{Y}} \pi_{|J|} \prod_{n \in J} \lambda_n$. Details are included in the supplementary materials (Suppl.). The computation complexity of this normalization depends on the eigen-decomposition of L. With the eigenvalues λ_n, we can compute the constant Z_G in polynomial time $O(N^2)$ with some slight change to the recursive algorithm [18, Algorithm

7], which calculates all the elementary symmetric polynomials $\sum_{|J|=k} \prod_{n \in J} \lambda_n$ for $k = 0, \cdots, N$ in $O(N^2)$ time. Therefore, the overall complexity of computing the normalization constant for GDPP is about the same as the complexity of normalizing an L-ensemble DPP (i.e., computing $\det(L + I)$).

Evaluation. With the normalization constant Z_G, we are ready to write out the probability of selecting a particular subset $y \subseteq \mathcal{Y}$ from the ground set by GDPP,

$$P_G(Y = y; L) = \frac{\pi_{|y|}}{Z_G} \det(L_y) \tag{12}$$

in which the concise form is due to the property of the elementary DPPs that $P_E(Y = y; J) = 0$ when $|y| \neq |J|$.

GDPP as a Mixture of k-DPPs. The GDPP expressed above has a close connection to the k-DPPs [17]. This is not surprising due to the definition of GDPP (cf. Eq. (11)). Indeed, GDPP can be exactly interpreted as a mixture of $N + 1$ k-DPPs $P_k(Y = y; L), k = 0, 1, \cdots, N$,

$$P_G(Y = y; L) = \frac{\pi_{|y|} \sum_{|J|=|y|} \prod_{n \in J} \lambda_n}{Z_G} P_{|y|}(Y = y; L)$$

if ***all the k-DPPs***, i.e., the mixture components, ***share the same L-ensemble kernel*** L as GDPP. If we introduce a new notation for the mixture weights, $p_k \triangleq \pi_k / Z_G \sum_{|J|=k} \prod_{n \in J} \lambda_n$, the GDPP can then be written as

$$P_G(Y; L) = \sum_{k=0}^{N} p_k P_k(Y; L). \tag{13}$$

Moreover, there is no necessity to adhere to the involved expression of p_k. Under some scenarios, directly playing with p_k may significantly ease the learning process. We will build a sequential model upon the GDPP of form (13) in the next section.

Exact Sampling. Following the interpretation of GDPP as a weighted combination of k-DPPs, we have the following decomposition of the probability:

$$P(Y|Y \sim G\mathrm{DPP}) = P(Y|Y \sim k - \mathrm{DPP}) P(k|k \sim G\mathrm{DPP}),$$

where, with a slight abuse of notation, we let $k \sim G$DPP denote the probability of sampling a k-DPP from GDPP. Therefore, we can employ a two-phase sampling procedure from the GDPP,

- Sample k from the discrete distribution $p = \{p_i\}_{i=0}^{N}$.
- Sample Y from k-DPP.

The supplementary materials present another sampling method via a Markov chain.

4.2 A Sequential Model of $\mathcal{G}$DPPs (Seq$\mathcal{G}$DPP)

In this section, we construct a sequential model of the generalized DPPs (Seq$\mathcal{G}$DPP) such that not only it models the temporal and diverse properties as SeqDPP does, but also allows users to specify the prior or constraint over the length of the video summary.

We partition a long video sequence $\mathcal{V}$ into T disjoint yet consecutive short segments $\bigcup_{t=1}^{T} \mathcal{V}_t = \mathcal{V}$. The main idea of Seq$\mathcal{G}$DPP is to adaptively distribute the expected length M_0 of the video summary to different video segments over each of which a $\mathcal{G}$DPP is defined. In particular, we replace the conditional DPPs in SeqDPP (cf. Eq. (4)) by $\mathcal{G}$DPPs,

$$P(X_t = x_t | X_{t-1} = x_{t-1}) \tag{14}$$

$$\triangleq P_{\mathcal{G}}(X_t = x_t; \Omega^t) = p^t_{|x_t|} P_{|x_t|}(X_t = x_t; \Omega^t), \tag{15}$$

where the last equality follows Eq. (13), and recall that the L-ensemble kernel Ω^t encodes the dependencies on the video frames/shots selected from the immediate past segment $x_{t-1} \subseteq \mathcal{V}_{t-1}$ (cf. Sect. 2, Eq. (5)). The discrete distribution $p^t = \{p^t_k\}$ is over all the possible sizes $\{k\}$ of the subsets at time step t.

We update p^t adaptively according to

$$p^t_k \propto \exp(-\alpha(k - \mu^t)^2), \tag{16}$$

where the mean $\mu^t \in [0, |\mathcal{V}_t|]$ is our belief about how many items should be selected from the current video segment $\mathcal{V}_t$ and the concentration factor $\alpha > 0$ tunes the confidence of the belief. When α approaches infinity, the $\mathcal{G}$DPP $P_{\mathcal{G}}(X_t; \Omega^t)$ degenerates to k-DPP and chooses exactly μ^t items into the video summary.

Our intuition for parameterizing the mean μ^t encompasses three pieces of information: the expected length M_0 over the overall video summary, number of items that have been selected into the summary up to the t-th time step, and the variety of the visual content in the current video segment $\mathcal{V}_t$. Specifically,

$$\mu^t \triangleq \frac{M_0 - \sum_{t'=1}^{t-1} |x_{t'}|}{T - t + 1} + w^T \phi(\mathcal{V}_t) \tag{17}$$

where the first term is the average number of items to be selected from each of the remaining video segments to make up an overall summary of length M_0, the second term $w^T \phi(\mathcal{V}_t)$ is an offset to the average number depending on the current video segment $\mathcal{V}_t$, and $\phi(\cdot)$ extracts a feature vector from the segment. We learn w from the training data—user annotated video summaries and their underlying videos. We expect that a visually homogeneous video segment gives rise to negative $w^T \phi(\mathcal{V}_t)$ such that less than the average number of items will be selected from it, and vice versa.

4.3 Learning and Inference

For the purpose of out-of-sample extension, we shall parameterize Seq$\mathcal{G}$DPP in such a way that, at time step t, it conditions on the corresponding video segment $\mathcal{V}_t$ and the selected shots $X_{t-1} = x_{t-1}$ from the immediate previous time step. We use a simple convex combination of D base GDPPs whose kernels are predefined over the video for the parameterization. Concretely, at each time step t,

$$\begin{aligned} P(X_t|x_{t-1}, \mathcal{V}_t) = P_G(X_t; \Omega^t, \mathcal{V}_t) &\triangleq \sum_{i=1}^{D} \beta_i P_G(X_t; \Omega^{t(i)}, \mathcal{V}_t) \\ &= \sum_{k=0}^{|\mathcal{V}_t|} p_k^t \sum_{i=1}^{D} \beta_i P_k(X_t; \Omega^{t(i)}, \mathcal{V}_t) \end{aligned} \tag{18}$$

where the L-ensemble kernels $\Omega^{t(i)}, i = 1, \cdots, D$ of the base GDPPs are derived from the corresponding kernels $L^{t(i)}$ of the conditional DPPs (Eq. (5)). We compute different Gaussian RBF kernels for $L^{t(i)}$ from the segment $\mathcal{V}_t$ and previously selected subset x_{t-1} by varying the bandwidths. The combination coefficients ($\beta_i \geq 0, \sum_i \beta_i = 1$) are learned from the training videos and summaries.

Consider a single training video $\mathcal{V} = \cup_{t=1}^{T} \mathcal{V}_t$ and its user summary $\{x_t \subseteq \mathcal{V}_t\}_{t=1}^{T}$ for the convenience of presentation. We learn Seq$G$DPP by maximizing the log-likelihood,

$$\begin{aligned} \mathcal{L} = \log \mathrm{Seq}G\mathrm{DPP} &= \sum_{t=1}^{T} \log P(X_t = x_t | x_{t-1}, \mathcal{V}_t) \\ = \sum_{t=1}^{T} \log p_{|x_t|}^t &+ \sum_{t=1}^{T} \log \left(\sum_{i=1}^{D} \beta_i P_{|x_t|}\left(X_t = x_t; \Omega_i^{t(i)}\right) \right). \end{aligned}$$

5 Experimental Setup and Results

In this section, we provide details on compiling an egocentric video summarization dataset, annotation process, and the employed evaluation procedure, followed by extensive comparison experiments on this dataset.

Dataset. While various video summarization datasets exist [7,11,33], we put consumer grade egocentric videos in our priority. They are often lengthy and carry a high level of redundancy, making summarization pressing need for the downstream applications. The UT Egocentric [19] dataset contains 4 videos each between 3–5 h long, covering activities such as driving, shopping, studying, etc. in uncontrolled environments. In this paper, we build our video summarization dataset by extending it with another 8 egocentric videos (on average over 6 h long each) from the social interactions dataset [8]. These videos are recorded using head-mounted cameras worn by individuals during their visits to Disney parks. Our efforts result in a dataset consisting of 12 long egocentric videos with a total duration of over 60 h.

Table 1. Some statistics about the lengths of the summaries generated by three annotators.

	User 1	User 2	User 3	Oracle
Min	79	74	45	74
Max	174	222	352	200
Avg.	105.75 ± 27.21	133.33 ± 54.04	177.92 ± 90.96	135.92 ± 45.99

User Summary Collection. We recruit three students to summarize the videos. The only instruction we give them is to operate on the 5-s video shot level. Namely, the full shot will be selected into the summary once any frame in the shot is chosen. Without any further constraints, the participants use their own preferences to summarize the videos at the granularities of their choice. Some statistics of Table 1 exhibit that the users have their own distinct preferences about the summary lengths.

Oracle Summaries. Supervised video summarization approaches are conventionally trained on one target summary per video. Having obtained 3 user summaries per video, we aggregate them into one *oracle summary* using a greedy algorithm that has been used in several previous works [9,30,31], and learn using them as the supervision. We leave the details of the greedy algorithm to the supplementary materials.

Features. We follow Zhang et al. [46] in extracting the features, i.e., using a pre-trained GoogleNet [37] to obtain the frame's pool5 activations and then aggregating them to a 1024-d feature representation for each shot of the video.

Evaluation. There has been a plethora of different metrics for evaluating the quality of video summaries including user studies [20,23], using low-level or pixel-level measurements to compare system summaries with user summaries [9,15,16,45,47], and temporal overlaps defined for two summaries [11,12,26,46]. We share the same opinion as [30,31,44] in that the evaluation of video summaries should take account of the high-level semantics the summaries convey.

To measure the quality of system summaries from the semantics perspective, Sharghi et al. [31] proposed to obtain dense shot-level concept annotations, termed as semantic vectors in which 1/0 indicates the presence/absence of a visual concept (e.g., Sky, Car, Tree, etc.). It is straightforward to measure the similarity between two shots using the intersection-over-union (IoU) of their concept vectors. For instance, if one shot is tagged by {Street,Tree,Sun} and the other by {Lady,Car,Street,Tree}, then the IoU is $2/5 = 0.4$. Having defined the similarity measure between shots, one can conveniently perform maximum weight matching on the bipartite graph, where the user and system summaries are placed on opposing sides of the graph.

Before collecting the per-shot concepts, we have to designate a good dictionary. We start with the dictionary of [31] and remove the concepts that do not

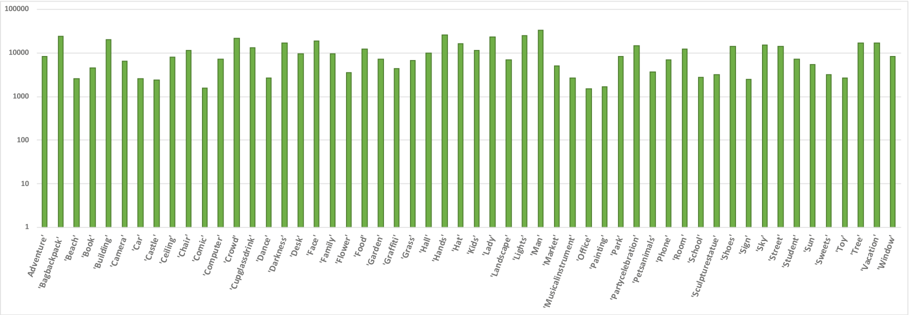


Fig. 1. Count of concept appearances in the collected annotations for the 12 videos.

appear frequently enough such as Boat and Ocean. Furthermore, we apply SentiBank detectors [3] (with over 1400 pre-trained classifiers) on the frames of the videos to make a list of visual concepts appearing commonly throughout the dataset. Next, by watching the videos, we select from this list the top candidates and append them into the final dictionary that includes 54 concepts (cf. Fig. 1).

Equipped with the dictionary of concepts, we uniformly sample 5 frames from each shot and ask Amazon Mechanical Turk workers to tag them with relevant concepts. The instruction here is that a concept must be selected if it appears in any of the 5 frames. We hire 3 Turkers per shot and pool their annotations by taking the union. On average, each shot is tagged with ∼11 concepts. This is significantly larger than the average of 4 tags/shot in Sharghi et al. [31], resulting in more reliable assessment upon evaluation. Amazon Mechnical Turk. Figure 1 shows the total number of each visual concept appeared in our dataset.

While the metric introduced in [31] compares summaries using the high-level concepts, it allows a shot in one summary to be matched with any shot in the other without any temporal restrictions. We modify this metric by applying a temporal filter on the pairwise similarities. We use two types of filters: (1) a Π (rectangular shaped) function and (2) a Gaussian function. The Π filter sets the similarities outside of a time range to zero, hence forcing the metric to match a shot to its temporally close candidates only. The Gaussian filter on the other hand applies a decaying factor on the matches far apart.

To evaluate a summary, we compare it to all 3 user-annotated summaries and average the scores. We report the performance by varying the filters' parameters (the temporal window size and the bandwidth in the Π and Gaussian filters, respectively). In addition, we compute the Area-Under-the-Curve (AUC) of the average F1-scores in Table 2. It is worth mentioning that setting the parameters of the filters to infinity results in the same metric defined by Sharghi et al. [31].

Data Split. In order to have a comprehensive assessment of the models, we employ a leave-one-out strategy. Therefore, we run 12 sets of experiments, each time leaving one video out for testing, two for validation (to tune hyperparameters), and the remaining 9 for training. We report the average results of the 12 rounds of experiments.

Large-Margin Training/Inference. Similar to the practices in seq2seq learning [27,40], we pre-train the models by maximizing the likelihood of user summaries using SGD. This finds a good initialization for the model, resulting in faster training process and better generalization to the test video. At the test time, we follow Eq. (6) to generate the system summary.

SeqGDPP Details. Given the features that are extracted using GoogleNet, we compute the Gaussian RBF kernels $\{L^{t(i)}\}_{i=1}^{D}$ over the video shots by varying the bandwidths $\sigma_i = 1.2^k\sigma_0$, where σ_0 is the median of all pairwise distances between the video shots. The base kernels $\{\Omega^{t(i)}\}$ for GDPPs are then computed through Eq. (5) such that they take account of the dependency between two adjacent time steps.

We also need to extract the feature vector $\phi(\mathcal{V}_t)$ to capture the variability in each video segment $\mathcal{V}_t$. In Eq. (17), we use such feature vector to help determine the mean of the distribution p over the possible subset sizes. Intuitively, larger subsets should be selected from segments with more frequent visual appearance changes. As such, we compute the standard deviation per feature dimension within the segment $\mathcal{V}_t$ for $\phi(\mathcal{V}_t)$.

There are three sets of parameters in Seq$G$DPP: α and w in the distribution over the subset size, and $\{\beta_i\}$ for the convex combination of some base GDPPs. We consider w and $\{\beta_i\}$ as model parameters to be learned by MLE or the large-margin algorithm and α as a hyper-parameter tuned according to the validation set.

Computational Cost Comparison. It takes about 28 s for SeqDPP to complete one epoch of the MLE training and about 4 s for Seq$G$DPP. The latter is faster because the kernel parameterization of Seq$G$DPP is less complex. The training time of either model doubles after we use the large-margin method to train it. This is not surprising because the large-margin method introduces extra cost for computing the margin. However, we find that this cost can be controlled in the following way. We first train the model (either SeqDPP or Seq$G$DPP) by the conventional MLE. After that, we fine-tune it by the large-margin method. By doing this, less than 10 epochs are required for the large-margin algorithm to converge.

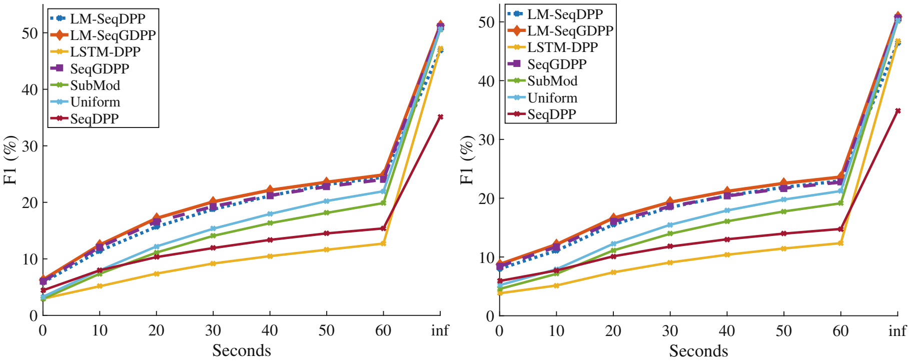
5.1 Quantitative Results and Analyses

In this section, we report quantitative results comparing our proposed models against various baselines:

- *Uniform.* As the name suggests, we sample shots with fixed step size from the video such that the generated summary has an equal length (the same number of shots) as the oracle summary.

Table 2. Comparison results for supervised video summarization (%). The AUCs are computed by the F1-score curves drawn in Fig. 2 until the 60 s mark. The blue and red colors group the base model and its large-margin version.

	AUC$_{\Pi}$	AUC$_{\text{Gaussian}}$
Uniform	12.33	12.36
SubMod [12]	11.20	11.12
SuperFrames [11]	11.46	11.28
LSTM-DPP [47]	7.38	7.36
SeqDPP [9]	9.71	9.56
LM-SeqDPP	15.05	14.69
SeqGDPP	15.29	14.86
LM-SeqGDPP	**15.87**	**15.43**



(a) Π temporal filter **(b)** Gaussian temporal filter

Fig. 2. Comparison results for supervised video summarization. The X axis represents the temporal filters' parameters. In the case of the Π filter, it indicates how far apart a match can be temporally (in terms of seconds), whereas in the Gaussian filter, it is the kernel bandwidth.

- *SubMod*. Gygli et al. [12] learn a convex combination of interestingness, representativeness, and uniformity from user summaries in a supervised manner. At the test time, given the expected summary length, which is the length of the oracle summary, the model generates the summary of that length.
- *SuperFrames*. In [11], Gygli et al. first segment the video into superframes and then measure their individual importance scores. Given the scores, the subsets that achieve the highest accumulative scores are considered the desired summary. Since a shot is 5-s long in our dataset, we skip the super-frame segmentation component. We train a neural network consisting of three fully-connected layers to measure each shot's importance score, and then choose the subsets with the highest accumulated scores as the summary.

- *LSTM-DPP.* In [46], Zhang et al. exploit LSTMs to model the temporal dependency between the shots of the video, and further use DPPs to enforce diversity in selecting important shots. Similar to previous baselines, this model has access to the expected summary length at the test time.
- *SeqDPP.* This is the original framework of Gong et al. [9]. Unlike other baselines, this model determines the summary length automatically.

The comparison results are shown in Table 2 and Fig. 2. There are some interesting observations as shown below.

(1) Comparing SeqDPP and the large-margin SeqDPP (denoted by LM-SeqDPP), we observe a significant performance boost thanks to the large-margin training algorithm. As illustrated in Fig. 2, the performance gap is consistently large throughout different filter parameters. Although both SeqDPP and LM-SeqDPP determine the summary lengths automatically, we find that the latter makes summaries that resemble the oracle summaries in terms of both length and semantic information conveyed.

(2) Comparing Seq$G$DPP to SeqDPP, for which users cannot tune the expected length of the summary, we can see that Seq$G$DPP significantly outperforms SeqDPP. This is not surprising since SeqDPP does not have a mechanism to take the user supplied summary length into account. As a result, the number of selected shots by SeqDPP is sometimes much less or more than the length of the user summary. Here both Seq$G$DPP and SeqDP are trained by MLE.

(3) The large-margin Seq$G$DPP (LM-SeqGDPP) performs slightly better than Seq$G$DPP, and it outperforms all the other methods. Nothing that both models generate summaries of the oracle lengths, the advantage of LM-SeqGDPP is soly due to that it selects the shots that better match the user summaries than Seq$G$DPP does.

(4) As described earlier, our refined evaluation scheme is a generalization of the bipartite matching of per-shot concepts [30]—if we set the filter parameters to infinity (hence no temporal restriction enforced by the filters), we can obtain the performance of the original metric. We can see from Fig. 2 that the relative orders of different methods remain about the same under different evaluation metrics but the refined one gives clearer and consistent margin between the methods. Hence, the AUC under the F1-score curve gives a more reliable quantitative comparison than the original metric (i.e., the rightmost points of the curves in Fig. 2).

6 Conclusion

In this work, we make twofold contribution towards improving the sequential determinantal point process (SeqDPP) models for supervised video summarization. We propose a large-margin training scheme that facilitates learning models more effectively by addressing the common problems in most seq2seq frameworks – exposure bias and loss-evaluation mismatch. Furthermore, we introduce a new probabilistic module, GDPP, which enables the resulting sequential model to accept priors about the expected summary length. Finally, we compile a large video summarization dataset consisting of 12 egocentric videos totalling over 60 h. We collecte 3 user-annotated summaries per video as well as dense concept annotations required for the evaluation. Experiments on this dataset verify the effectiveness of our large-margin training algorithm as well as the sequential *G*DPP model.

Acknowledgements. This work was supported in part by NSF IIS 1741431 & 1566511, gifts from Adobe, and gift GPUs from NVIDIA.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Borodin, A., Rains, E.M.: Eynard-Mehta theorem, Schur process, and their pfaffian analogs. J. Stat. Phys. **121**(3), 291–317 (2005)
3. Borth, D., Ji, R., Chen, T., Breuel, T., Chang, S.F.: Large-scale visual sentiment ontology and detectors using adjective noun pairs. In: Proceedings of the 21st ACM International Conference on Multimedia, pp. 223–232. ACM (2013)
4. Chu, W.S., Song, Y., Jaimes, A.: Video co-summarization: video summarization by visual co-occurrence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3584–3592 (2015)
5. Collins, M., Roark, B.: Incremental parsing with the perceptron algorithm. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 111. Association for Computational Linguistics (2004)
6. Daumé, H., Langford, J., Marcu, D.: Search-based structured prediction. Mach. Learn. **75**(3), 297–325 (2009)
7. 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. Pattern Recog. Lett. **32**(1), 56–68 (2011)
8. Fathi, A., Hodgins, J.K., Rehg, J.M.: Social interactions: a first-person perspective. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1226–1233. IEEE (2012)
9. Gong, B., Chao, W.L., Grauman, K., Sha, F.: Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems, pp. 2069–2077 (2014)
10. Gong, B., Chao, W., Grauman, K., Sha, F.: Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems (NIPS), pp. 2069–2077 (2014)
11. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool, L.: Creating summaries from user videos. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 505–520. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_33
12. Gygli, M., Grabner, H., Van Gool, L.: Video summarization by learning submodular mixtures of objectives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3090–3098 (2015)
13. Hirsch, R.: Seizing the Light: A Social and Aesthetic History of Photography. Taylor & Francis, Routledge (2017)

14. Hough, J.B., Krishnapur, M., Peres, Y., Virág, B.: Determinantal processes and independence. Probab. Surv. **3**, 206–229 (2006)
15. Khosla, A., Hamid, R., Lin, C.J., Sundaresan, N.: Large-scale video summarization using web-image priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2698–2705 (2013)
16. Kim, G., Sigal, L., Xing, E.P.: Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4225–4232 (2014)
17. Kulesza, A., Taskar, B.: k-DPPs: fixed-size determinantal point processes. In: Proceedings of the 28th International Conference on Machine Learning (ICML), pp. 1193–1200 (2011)
18. Kulesza, A., Taskar, B.: Determinantal point processes for machine learning. Found. Trends® Mach. Learn. **5**(2–3), 123–286 (2012)
19. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1346–1353. IEEE (2012)
20. Lee, Y.J., Grauman, K.: Predicting important objects for egocentric video summarization. Int. J. Comput. Vis. **114**(1), 38–55 (2015)
21. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL 2004 Workshop, Barcelona, Spain, vol. 8 (2004)
22. Liu, W., Mei, T., Zhang, Y., Che, C., Luo, J.: Multi-task deep visual-semantic embedding for video thumbnail selection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3707–3715 (2015)
23. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2714–2721 (2013)
24. Obile, W.: Ericsson mobility report (2016)
25. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
26. Potapov, D., Douze, M., Harchaoui, Z., Schmid, C.: Category-specific video summarization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 540–555. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_35
27. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732 (2015)
28. Ross, S., Gordon, G.J., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: International Conference on Artificial Intelligence and Statistics, pp. 627–635 (2011)
29. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: AAAI, pp. 3776–3784 (2016)
30. Sharghi, A., Gong, B., Shah, M.: Query-focused extractive video summarization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 3–19. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_1
31. Sharghi, A., Laurel, J.S., Gong, B.: Query-focused video summarization: dataset, evaluation, and a memory network based approach. arXiv preprint arXiv:1707.04960 (2017)

32. Shen, S., et al.: Minimum risk training for neural machine translation. arXiv preprint arXiv:1512.02433 (2015)
33. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: TVSum: summarizing web videos using titles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5179–5187 (2015)
34. Sun, M., Farhadi, A., Seitz, S.: Ranking domain-specific highlights by analyzing edited videos. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 787–802. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_51
35. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 1017–1024 (2011)
36. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
37. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
38. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence-video to text. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4534–4542 (2015)
39. Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., Hinton, G.: Grammar as a foreign language. In: Advances in Neural Information Processing Systems, pp. 2773–2781 (2015)
40. Wiseman, S., Rush, A.M.: Sequence-to-sequence learning as beam-search optimization. arXiv preprint arXiv:1606.02960 (2016)
41. Wolf, W.: Key frame selection by motion analysis. In: 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1996, vol. 2, pp. 1228–1231. IEEE (1996)
42. Xiong, B., Grauman, K.: Detecting snap points in egocentric video with a web photo prior. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 282–298. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_19
43. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057 (2015)
44. Yeung, S., Fathi, A., Fei-Fei, L.: VideoSET: video summary evaluation through text. arXiv preprint arXiv:1406.5824 (2014)
45. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: 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)
46. Zhang, K., Chao, W.-L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 766–782. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_47
47. Zhao, B., Xing, E.P.: Quasi real-time summarization for consumer videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2513–2520 (2014)