2019

Visual-Textual Video Synopsis Generation

Aidean Sharghi Karganroodi
University of Central Florida

VISUAL-TEXTUAL VIDEO SYNOPSIS GENERATION

by

AIDEAN SHARGHI KARGANROODI
B.S. Sharif University of Technology, 2013

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2019

Major Professor: Mubarak Shah

ABSTRACT

In this dissertation we tackle the problem of automatic video summarization. Automatic summarization techniques enable faster browsing and indexing of large video databases. However, due to the inherent subjectivity of the task, no single video summarizer fits all users unless it adapts to individual user's needs. To address this issue, we introduce a fresh view on the task called "Query-focused" extractive video summarization. We develop a supervised model that takes as input a video and user's preference in form of a query, and creates a summary video by selecting key shots from the original video. We model the problem as subset selection via determinantal point process (DPP), a stochastic point process that assigns a probability value to each subset of any given set. Next, we develop a second model that exploits capabilities of memory networks in the framework and concomitantly reduces the level of supervision required to train the model. To automatically evaluate system summaries, we contend that a good metric for video summarization should focus on the semantic information that humans can perceive rather than the visual features or temporal overlaps. To this end, we collect dense per-video-shot concept annotations, compile a new dataset, and suggest an efficient evaluation method defined upon the concept annotations. To enable better summarization of videos, we improve the sequential DPP in two folds. In terms of learning, we propose a large-margin algorithm to address the exposure bias that is common in many sequence to sequence learning methods. In terms of modeling, we integrate a new probabilistic distribution into SeqDPP, the resulting model accepts user input about the expected length of the summary. We conclude this dissertation by developing a framework to generate textual synopsis for a video, thus, enabling users to quickly browse a large video database without watching the videos.

EXTENDED ABSTRACT

Every minute, 300 hours of video is being uploaded to YouTube. With this tremendous amount of available video, it is impossible to watch them all. As a result of the “big video data", intelligent algorithms for automatic video summarization have (re-)emerged as a pressing need. However, one of the main obstacles for the research in video summarization is the subjective-ness — users have diverse preferences over the summaries. The subjectiveness causes at least two problems. First, no single video summarizer fits all users unless it interacts with and adapts to the individual users. Second, it is very challenging to evaluate the performance of a video summarizer. To tackle the first challenge, we develop a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP), for **query-focused** extractive video summarization. Given a user query and a long video sequence, our algorithm returns a summary by selecting key shots from the video. It has two layers of random variables, each of which serves for subset selection from a ground set of video shots. The first layer is mainly used to select the shots relevant to the user queries, and the second layer models the importance of the shots in the context of the videos. We condition the second layer on the first layer so that we can automatically balance the two strengths by learning from user labeled summaries.

We contend that the pursuit of new algorithms for video summarization has actually left the basic problem — how to benchmark different video summarizers — under-explored. User study is too time-consuming to compare different approaches and their variations at large scale. To automate the evaluation procedure, on one end, a system generated summary has to select the same key units (frame or shot) as the users do in order to be counted as a good one, although many key units could be visually similar to the user labeled ones. On the other end, pixels and low-level features are used to compare the system and user summaries, whereas it is unclear what features and distance metrics match users’ criteria. We argue that a good evaluation metric for video summarization

should focus on the semantic information that humans can perceive rather than the visual features or temporal overlaps. To this end, we collect dense per-video-shot concept annotations, compile a new dataset, and suggest an efficient and automatic evaluation method defined upon the concept annotations. In addition, we propose a memory network parameterized SeqDPP for tackling the query-focused video summarization. Unlike the hierarchical model, this framework does not rely on the costly user supervision about which queried concept appears in which video shot or any pre-trained concept detectors. Instead, we use the memory network to implicitly attend the user query about the video onto different frames within each shot.

Next, to address the limitations of SeqDPPs, we make two-pronged contribution towards improving models to more effectively learn better video summarizers. In terms of learning, we propose a large-margin algorithm to address the SeqDPP's exposure bias; a mismatch issue in many sequence to sequence (seq2seq) learning methods. 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 between the training phase and the inference. To tackle these issues, we adapt the *Large-Margin* algorithm originally derived for training LSTMs to the SeqDPPs. The main idea is to alleviate the exposure bias by incorporating inference techniques of the test time into the objective function used for training. Meanwhile, we add to the large-margin formulation a multiplicative reward term that is directly related to the evaluation metrics to mitigate the loss-evaluation mismatch. 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. To this end, we propose a generalized DPP (*G*DPP) in which an arbitrary prior distribution can be imposed over the sizes of subsets of video shots. As a result, both vanilla DPP and k-DPP can be considered as special instances of *G*DPP. Moreover, we can conveniently substitute the (conditional) DPPs in SeqDPP by *G*DPP. 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 numbers of video shots from corresponding segments.

Finally, we argue that it is highly advantageous to model video summarization as a natural language (text) generation task. More specifically, given a video, our approach returns a short textual summary to the user. This enables users to quickly index a large video database without watching them. To achieve this goal, initially, video is divided into non-overlapping shots and their visual features are extracted. An LSTM-based video caption generation network is employed to generate a textual description, in form of a natural language sentence, for each shot in the video given its feature representations. Each generated sentence is passed through a module that assigns to it a naturalness score based on its resemblance to sentences written by human subjects. In parallel, another module processes all the generated sentences in temporal order, assigning a significance score to each based on their importance in the context of the video. Next, summary-level impact of each sentence, represented by a scalar, is formulated as the product of its naturalness and significance scores. In other words, we evaluate worthiness of a sentence based on its individual quality as well as the importance of the event that it is describing in the context of the video. The summary-level impact values in the temporal order form a time series where the peaks correspond to locally (in time) important events in the video. We put the sentences with peak impact values in a document in temporal order and return this text synopsis to the user.

To my beloved parents

my dear brother

and my beautiful girlfriend

whom I owe everything I am to

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Mubarak Shah for his great support throughout these years. Also, I would like to thank Prof. Niels Niels Da Vitoria Lobo, Prof. Nazanin Rahnavard, and Prof. George Atia for being a part of my committee. Finally, I would like to thank all of the past and present members of Center for Research in Computer Vision (CRCV), Cherry Place, Tonya LaPrarie, Kenneth Davis, Enrique Ortiz, Amir Roshan Zamir, Afshin Dehghan, Shervin Ardeshir, Shayan Modiri, Amir Mazaheri, Aisha Orooj Khan, Krishna Regmi, Dong Zhang, Yonatan Tariku, Khurram Soomro, Waqas Sultani, and Sarfaraz Hussein for their help and the memories.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

As video acquisition devices become more and more widespread, the corpus of available video content is exponentially growing. With the emergence of video logging (vlog) as a phenomenon, the rise in number of security cameras installed all around the glob, aerial videography using UAVs and drones, and the increase in utilizing body-worn and dash cameras by police officers everywhere, it has become extremely challenging to browse, index, or extract information from this rich resource. Hence, effective video summarization frameworks are now necessity rather than luxury. However, one of the main obstacles for research on video summarization is the subjectiveness factor — users have diverse preferences over the summaries. The subjectiveness causes at least two problems. First, no single video summarizer satisfies all users' needs unless it interacts with and adapts to the individual users. Second, it is very challenging to evaluate the performance of a video summarizer.

This dissertation contributes to video summarization by proposing: (1) Sequential and Hierarchical Determinantal Point Process (SH-DPP) that models user input, or more precisely user intentions, in the summarization process via a two layer structure, (2) a memory network parameterized sequential determinantal point process in order to attend the user query onto different video frames and shots, (3) a large-margin learning objective to alleviate the exposure bias and loss-evaluation mismatch, (4) an efficient and automatic evaluation method, (5) a new probabilistic distribution, generalized DPP (*G*DPP), such that when it is integrated into SeqDPP, the resulting model accepts user input about the expected length of the summary, (6) a novel framework to generate text synopsis for a given video.

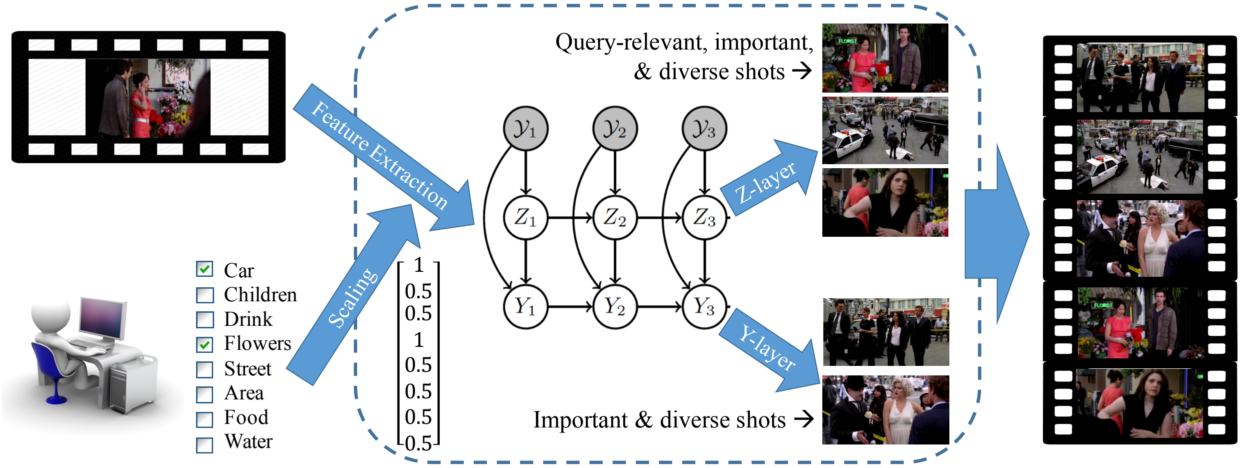
1.1 Query-Focused Extractive Video Summarization

In this dissertation, we focus on *extractive* video summarization, which generates a concise summary of a video by selecting from it the key frames or shots. The key frames or shots are expected to be 1) individually important—otherwise they should not be selected, and 2) collectively diverse—otherwise one could remove some of them without losing much information. These two principles are employed in most of the existing works on extractive video summarization [5], and yet implemented by different decision choices. Some earlier works define the importance of key frames by low-level appearance and/or motion cues [6, 7, 8, 9, 10, 11]. Contextual information of a key frame is often modeled by graphs [12, 13, 14]. We note that the system developers play a vital role in this collection of works; most decisions on how to measure the importance and diversity are handcrafted by the system developers using low-level cues.

From the more recent works, we see a paradigm shift: more high-level supervised information is introduced to video summarization than ever before. Rich Web images and videos provide (weakly) supervised priors for defining **user-oriented** importance of the visual content in a video [15, 16, 17, 18]. For instance, the CAR images on the Web reveal the canonical views of the cars liked by average users, which should thus be given special attention in video summarization. The texts associated with videos are undoubtedly good sources for inferring the semantic importance of video frames [19, 20]. Category-specific and domain-specific video summarization approaches are developed in [21, 22]. Some other high-level factors include gaze [23], interestingness [24], influence [25], tracking of salient objects [26, 27], and so forth.

What are the advantages of leveraging high-level supervised information in video summarization over merely low-level cues? We believe the main advantage is that the system developers are able to better infer the system **users'** needs. After all, video summarization is a subjective process. When compared to designing the system from the experts' own intuitions, it is more desirable to design a system based on the crowd or average users such that the system's states approach the users' internal states, which are often semantic and high-level.

What is the best supervision for a video summarization system? We have seen many types of supervision used in the above-mentioned works, such as Web images, texts, categories, etc. However, we argue that the best supervision, for the purpose of developing video summarization approaches, is the video summaries directly provided by the **users/annotators**. In [1], which is the first supervised video summarization work as far as we know, Gong et al. showed that there exists a high inter-annotator agreement in the summaries of the same videos given by distinct users. They proposed a supervised video summarization model, sequential determinantal point process (seqDPP), and train seqDPP by the "oracle" summaries that agree the most among different user summaries. Gygli et al. developed another supervised approach to video summarization by learning submodular functions from the user summaries [28].



(a) Input: Video & Query **(b) Algorithm**: Sequential & Hierarchical Determinantal Point Process (SH-DPP) **(c) Output**: Summary

Figure 1.1: Query-focused video summarization and our approach to this problem. The input to the model is a video and user's preferences. The model follows a hierarchical architecture in which the top layer summarizes the video with respect to the query, and the bottom layer is responsible for summarizing what remains of the video.

From the low-level visual and motion cues to the high-level (indirect) supervised information, and to the (direct) supervised user summaries, video summarization is becoming more and more **user-oriented**. Though the two principles, importance and diversity, remain the same, the detailed implementation choices have significantly shifted from the system developers to the system users; users can essentially teach the system how to summarize videos in [1, 28].

We aim to further advance the user-oriented video summarization by modeling user input, or more precisely user intentions, in the summarization process. Figure 1.1 illustrates our main idea. We name it query-focused (extractive) video summarization, in accordance with the query-focused document summarization [29] in NLP. Here, a query refers to one or more concepts (e.g., CAR, FLOWERS) that are both user-nameable and machine-detectable. More generic queries are left for future work.

Towards the goal of query-focused video summarization, we develop a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP). It has two layers of random variables, each of which serves for subset selection from a ground set of video shots (see Figure 1.2). The first layer is mainly used to select the shots relevant to the user queries, and the second layer models the importance of the shots in the context of the videos. We condition the second layer on the first layer so that we can automatically balance the two strengths by learning from user labeled summaries. The determinantal point process (DPP) [30] is employed to account for the diversity of the summary.

A key feature in our work is that the decision to include a video shot in the summary is jointly dependent on the shot's relevance to the query and representativeness in the video. Instead of handcrafting any decision criteria, we use the SH-DPP probabilistic model to automatically learn from the user summaries (and the corresponding user queries and video sequences). In a sharp contrast to [1, 28] which model average users, our work closely tracks individual users' intentions from their input queries, and thus has greater potential to satisfy various user needs: distinct personal preferences (e.g., a patient user prefers more detailed and lengthy summaries than an impatient user), different interests over time even about the same video (e.g., a party versus a particular person in the party), etc. Finally, we note that our work is especially useful for search engines to produce snippets of videos.

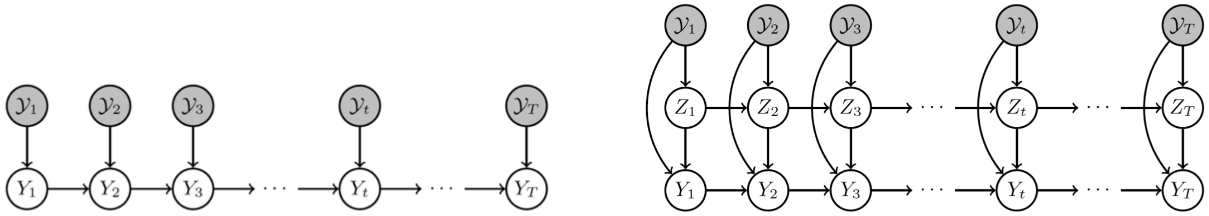


Figure 1.2: The graphical models of seqDPP [1] (left) and our SH-DPP (right). SH-DPP has an extra layer of sequential DPPs compared to the standard seqDPP.

Our first contribution is on the query-focused video summarization. Querying videos is not only an appealing functionality to the users but also an effective communication channel for the system to capture a user's intention. Besides, we develop a novel probabilistic model, SH-DPP. Similar to the sequential DPP (seqDPP) [1], SH-DPP is efficient in modeling extremely lengthy videos and capable of producing summaries on the fly. Additionally, SH-DPP explicitly accounts for the user input queries. Extensive experiments on the UT Egocentric [31] and TV episodes [32] datasets verify the effectiveness of SH-DPP. To our knowledge, our work is the first on query-focused video summarization.

While this may be a promising direction to personalize video summarizers, our study was conducted on the datasets originally collected for conventional generic video summarization [33, 2]. It was unclear whether the real users would generate distinct summaries for different queries, and if yes, how much the query-focused summaries differ from each other. Additionally, the hierarchical model relies on the costly user supervision about which queried concept appears in which video shot or any pre-trained concept detectors. To alleviate these issues, we compile the first query-focused video summarization dataset and densely annotate it with concepts for evaluation purposes. Furthermore, we use a memory network to implicitly attend the user query about the video onto different frames within each shot, discussed in the next Section.

1.2 A Memory Network Based Approach, Dataset, and Evaluation

In this section, we explore more thoroughly the query-focused video summarization and build a new dataset particularly designed for it. While we collect the user annotations, we encounter the challenge of how to define a good evaluation metric to contrast system generated summaries to user labeled ones — the problem we mentioned above, that is due to individual users' subjectiveness over the video summaries.

We contend that the pursuit of new algorithms for video summarization has actually left the basic problem — how to benchmark different video summarizers — under-explored. User study [26, 25] is too time-consuming to compare different approaches and their variations at large scales. To automate the evaluation procedure, on one end, a system generated summary has to select the same key units (frame or shot) as the users do in order to be counted as a good one [18, 19, 23], although many key units could be visually similar to the user labeled ones. On the other end, pixels and low-level features are used to compare the system and user summaries [1, 15, 16, 34, 35], whereas it is unclear what features and distance metrics match users' criteria. Some works strive to find a balance between the two extremes, e.g., using the temporal overlap between two summaries to define the evaluation metrics [24, 28, 21, 36]. However, all such metrics are derived from either the temporal or visual representations of the videos, without explicitly encoding how humans perceive the information — after all, the system generated summaries are meant to deliver similar information to the users as those directly labeled by the users.

In terms of defining a better measure that closely tracks what humans can perceive from the video summaries, we share the same opinion that of Yeung et al. [2]: it is key to evaluate how well a system summary is able to retain the semantic information, as opposed to the visual quantities, of the user supplied video summaries. Arguably, the semantic information is best expressed by concepts (e.g., objects, places, people, etc.) that represent the fundamental characteristics of what we see in the video.

Therefore, we collect dense per-video-shot concept annotations for our dataset. In other words, we represent the semantic information in each video shot by a binary semantic vector, in which the 1's indicate the presence of corresponding concepts in the shot. We suggest a new evaluation metric for the query-focused (and generic) video summarization based on these semantic vector representations of the video shots.

In addition, we propose a memory network [37] parameterized sequential determinantal point process [1] for tackling the query-focused video summarization. Unlike the hierarchical model we introduced earlier, this approach does not rely on the costly user supervision about which queried concept appears in which video shot or any pre-trained concept detectors. Instead, we use the memory network to implicitly attend the user query about the video onto different frames within each shot. Extensive experiments verify the effectiveness of our approach.

In the next Chapter, we make a two-pronged contribution towards improving these models to more effectively learn better video summarizers. 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.

1.3 Generalized DPPs and a Large-Margin Objective

This section is in the vein of supervised video summarization based on a determinantal point process (DPP) [30]. 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 [1] for handling sequences, it has been employed in a rich line of recent works on video summarization [38, 39].

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 [40, 41, 42, 43, 44]. 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 [45] between the training phase and the inference. To tackle these issues, we adapt the *Large-Margin* algorithm originally derived for training LSTMs [46] to the SeqDPPs. The main idea is to alleviate the exposure bias by incorporating inference techniques of the test time into the objective function used for training. Meanwhile, we add to the large-margin formulation a multiplicative reward term that is directly 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 (*G*DPP) in which an arbitrary prior distribution can be imposed over the sizes of subsets of video shots. As a result, both vanilla DPP and k-DPP [47] can be considered as special instances of *G*DPP. Moreover, we can conveniently substitute the (conditional) DPPs in SeqDPP by *G*DPP. When a user specifies an expected length of summary, we dynamically allocate it to different segments of the video and then choose the right numbers of video shots from corresponding segments.

We conduct extensive experiments to verify the improved techniques for supervised video summarization. First of all, we significantly extend the UTE dataset [33] and its annotations of video summaries and per-shot concepts [39] by another eight egocentric videos [48]. Following the protocol described in [39], 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 *G*DPP but also the existing SeqDPP models.

1.4 Text Synopsis Generation for Videos

Traditionally, video summarization techniques aim to create a short video summary that consists of a diverse set of important frames or shots. Even though such summaries are significantly shorter than the original videos, system users must watch these summaries to extract information from them. While this might not impose a problem when the database is small, however, the ultimate goal of video summarization, that is to effectively browse large databases, remains a challenge. An ideal video summarization framework must enable users to quickly go through a large amount of visual data without actually watching them.

In this section, we develop a novel framework to generate text synopsis for a given video. Similar to conventional methods, the input to our model is a video, however the main output in our system is a short textual summary. This summary consists of several sentences that are chosen based on their correctness in describing the video as well as their significance in conveying important information about the video when considered together. In addition to the textual summary, video shots corresponding to the sentences in textual summaries can be easily retrieved, temporally sorted and stitched together to generate a visual summary similar to the output of conventional methods.

To achieve this goal, initially, the video is divided into non-overlapping shots and their visual features are extracted. An LSTM-based video caption generation network is employed to generate a textual description, in the form of a natural language sentence, for each shot in the video given its feature representations. Each generated sentence is passed through a module that assigns to it a correctness score that measures how well it describes the visual content of the video shot. In parallel, another module processes all the generated sentences in temporal order, assigning a significance score to each based on their importance in the context of the video. Next, summary-level impact of each sentence, represented by a scalar, is formulated as the product of its correctness and significance scores. In other words, we evaluate worthiness of a sentence based on its individual quality as well as the importance of the event that it is describing in the context of the video. The summary-level impact values in the temporal order form a time series where the peaks correspond to locally (in time) important events in the video. We put the sentences with peak impact values in a document in temporal order and return this text synopsis to the user.

There are several advantages to our framework over conventional methods. Firstly, our model exploits semantic information in its most expressive form, i.e., in natural language domain, which leads to a rather significant improvement in the quality of the generated summaries. More importantly, since the summaries are in form of text, users can browse a large number of videos as quickly as possible without watching them. Furthermore, one can conveniently search through a large video database via text queries. Textual summaries generated by our model are compared to groundtruth summaries in the text domain via the ROUGE [3] metric, a well-established metric in automatic document summary evaluation.

1.5 Summary

In this Chapter, we discussed in Section 1.1 the need for personalizing video summarization such that they better fit each individual user's needs and introduced Sequential and Hierarchical Determinantal Point Process (SH-DPP that models user input, or more precisely user intentions, in the summarization process via a two layer structure). In Section 1.2, we explained the challenge on how to define a good evaluation metric to contrast system generated summaries to user labeled ones, and moreover, a novel memory network based approach. In Section 1.3, we explain the exposure bias and loss-evaluation mismatch issues, common in many sequence to sequence learning algorithms and the limitation of SeqDPP in accepting expected summary length and how to tackle it. Finally, in Section 1.4, we argue the advantages of producing textual synopses for videos in enabling faster browsing of large video databases.

1.6 Organization

In Chapter 2, we review existing literature on video summarization and Determinantal Point Processes. In Chapter 3, we present our proposed SH-DPP to model the user input in form of queries in the summarization process. In Chapter 4, we explain in details of compiling the first query-focused video summarization dataset and our novel memory network based SeqDPP approach to better summarize the videos. In Chapter 5, we reformulate the training objective for SeqDPP based models to more effectively learn better summarizers, and furthermore, we develop a novel probabilistic distribution such that, when it is integrated into SeqDPP, the resulting model accepts user input about the expected length of the summary. Finally, in Chapter 6, we develop a novel framework to generate text synopsis for a given video. Similar to conventional methods, the input to our model is a video, however the main output in our system is a short textual summary. This summary consists of several sentences that are chosen based on their correctness as well as their significance in conveying important information about the video when considered together.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we comprehensively study the literature on video summarization in Section 2.1. In Section 2.6, we formally introduce determinantal point processes as it is a building block in our frameworks.

2.1 Video Summarization

Video summarization is an inherently complex task. On an intuitive level, video summarization techniques aim to create a significantly shorter version of a given video such that the user can gain most information as quickly as possible. However, depending on the content, style, lengths, etc. of the videos, the criteria for what constitutes a good summary can change drastically. In addition to that, since the video content has grown significantly, a wide variety of methods have been developed by researchers.

Existing methods can be categorized from several perspectives such as the type(s) of features they use, level of supervision in the models, and the criteria for identifying key frames/shots. Early works define the importance of key frames by low-level appearance and/or motion cues [6, 7, 8, 9, 10, 11]. Contextual information of a key frame is often modeled by graphs [12, 13, 14]. We note that the system developers play a vital role in this cohort of works; most decisions on how to measure the importance and diversity are handcrafted by the system developers using the low-level cues.

From the more recent works, we see a paradigm shift in some sort: more high-level supervised information is introduced to video summarization than ever before. Rich Web images and videos provide (weakly) supervised priors for defining importance of the visual content in a video [15, 16, 17, 18]. The texts associated with videos are undoubtedly good sources for inferring the semantic importance of video frames [19, 20]. Category-specific and domain-specific video summarization approaches are developed in [21, 22]. Some other high-level factors include gaze [23], interestingness [24], influence [25], tracking of salient objects [26, 27], and so forth.

2.2 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 [15, 16, 17, 18], weakly supervised web image and video priors help define visual importance. Captions associated with videos are used by [19, 20] to infer semantic importance. Finally, many frameworks (e.g., [36, 1, 28]) learn a summarizer directly from user-annotated summaries.

In [1], which is the first supervised video summarization work as far as we know, Gong et al. showed that there exists a high inter-annotator agreement in the summaries of the same videos given by distinct users. They proposed a supervised video summarization model, sequential determinantal point process (seqDPP), and train seqDPP by the “oracle” summaries that agree the most among different user summaries. 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.

2.3 Personalizing Video Summaries

Interactive video summarization shares some spirits with our query-focused video summarization. The system in [49] allows users to interactively select some video shots to the summary while the system summarizes the remaining video. In contrast, in our system the users can use concept-based queries to influence the summaries without actually watching the videos. Besides, our approach is supervised and trained by user annotations, not handcrafted by the system developers. There are some other works involving users in the summarization related tasks, such as thumbnail selection [20] and storyline-based video representation [50]. Our work instead involves user input in the video summarization.

In respect to the recent progress, we aim to further advance the user-oriented video summarization by modeling user input, or more precisely user intentions, in the summarization process. We name it query-focused (extractive) video summarization, in accordance with the query-focused document summarization [29] in NLP, a long-standing track in the Text Retrieval Conference (`http://trec.nist.gov/`) and the Document Understanding Conference (DUC) (`http://duc.nist.gov/`). In DUC 2005 for example, participants were asked to summarize a cluster of documents given a user's query describing the information needs. Some representative approaches to this problem include BAYESUM [51], FASTSUM [52], and log-likelihood based method [53] among others. Behind the vast research in this topic is the strong motivations by popular search engines and human-machine interactions. However, the counterpart in vision, query-focused video summarization, has not been well formulated yet.

We propose two frameworks to perform query-focused video summarization in this dissertation.

Our first approach is purely based on DPPs while in second model, we propose a memory network [37] parameterized sequential determinantal point process [1] for tackling the query-focused video summarization. Unlike our hierarchical model, this approach does not rely on the costly user supervision about which queried concept appears in which video shot or any pre-trained concept detectors. Instead, we use the memory network to implicitly attend the user query about the video onto different frames within each shot. Memory networks [42, 37, 54, 55, 56] are versatile in modeling the attention scheme in neural networks. They are widely used to address question answering and visual question answering [57]. The query focusing in our summarization task is analogous to attending questions to the "facts" in the previous works, but the facts in our context are temporal video sequences. Moreover, we lay a sequential determinantal point process [1] on top of the memory network in order to promote diversity in the summaries.

2.4 Video Summarization from Sequence-to-Sequence Learning Perspective

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 [42] and text generation applications (e.g., for image and video captioning [58, 59]).

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 [3] and BLEU [60]. Inevitably, such mismatches degreade the overall performance of a seq2seq model not because of the model's own capacity but due to the suboptimal training algorithms.

We aim to improve these models from the perspective of learning strategy. In terms of learning, we propose a large-margin algorithm to address the SeqDPP's exposure bias problem explained below. Exposure bias is in fact a common issue in many sequence-to-sequence learning methods [40, 41, 42, 43, 44]. 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 [45] between the training phase and the inference.

2.5 Video Summarization in Text Domain

Traditionally, video summarization techniques aim to create a short video summary that consists of a diverse set of important frames or shots. Even though such summaries are significantly shorter than the original videos, system users must watch these summaries to extract information from them. To develop a framework to generate text synopsis for a given video, one needs to generate a textual description, in form of a natural language, for the video. This area of research is known in the community as video caption generation.

Video caption generation covers a few slightly different research topics. In single-sentence captioning, as the name suggests, the objective is to produce a single description for a video. Early efforts such as [61, 62], are template-based, that is to fill in part-of-speech tags with the actions, places, and objects. After the re-emergence of neural networks, due to their superior performance, several methods following an encoder-decoder structure were proposed [63, 64, 65, 66, 58]. Such frameworks use recurrent neural networks (RNNs) such as LSTM and GRU to first encode the video, and then generate a caption for it using a decoding RNN. More recent works take advantage of attention mechanism to boost the performance [59]. In dense video captioning, the goal is to produce multiple captions. These approaches [67, 68, 69] usually achieve this goal via a proposal unit that targets different segments in the video. Finally, the caption generation module generates a single sentence for each segment. In video captioning with a paragraph, the focus is on giving a more detailed description of the video by generating several "semantically-fluent" sentences [70, 71]. This is similar to natural language generation except that the data in this type of caption generation is in form of videos.

Once we transform the video into text domain, the extracted knowledge needs to be summarized, which shares about the same opinion in document summarization works. According to Radev et al. [72], a summary is text produced from one or more text documents that preserves the important information of the original texts and is usually significantly shorter than half of the original documents. Automatic text summarization methods can be divided into *extractive* [73, 74, 75, 76] and *abstractive* [77, 78, 79, 80], depending on whether summarization is done via a sentence selection or generation process. Extractive document summarization and video summarization highly resemble each other except that the input and output domains are different.

While the use of natural language for video summarization is not new [81, 82], to the best of our knowledge, we are the first to generate as much text knowledge as we can about the video, and then solve the problem as a document summarization under severe noise.

In the next Section, we formally introduce Determinantal Point Processes and its sequential variant to facilitate reading the rest of this Dissertation. SeqDPP is a building block in the architectures developed in the next three Chapters.

2.6 Determinantal Point Process

A discrete DPP [30, 83] 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 $\mathcal{Y} = \{1, ..., N\}$, one can define $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, 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 $\boldsymbol{y} \subseteq \mathcal{Y}$ the following holds:

$$P(\boldsymbol{y} \subseteq Y; \boldsymbol{K}) = \det(\boldsymbol{K}_{\boldsymbol{y}}), \tag{2.1}$$

where $\boldsymbol{K}_{\boldsymbol{y}}$ is the squared sub-kernel of $\boldsymbol{K}$ with rows and columns indexed by the elements in $\boldsymbol{y}$, and $\det(.)$ is the determinant function. $\boldsymbol{K}$ is referred to as the marginal kernel since one can compute the probability of any subset $\boldsymbol{y}$ being included in $\mathcal{Y}$. It is the property of the determinant that promotes diversity: in order to have a high probability $P(i, j \in Y; \boldsymbol{K}) = \boldsymbol{K}_{ii}\boldsymbol{K}_{jj} - \boldsymbol{K}_{ij}^2$, the per-element importance terms $\boldsymbol{K}_{ii}$ and $\boldsymbol{K}_{jj}$ 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 $\boldsymbol{L} = \boldsymbol{K}(\boldsymbol{I} - \boldsymbol{K})^{-1}$ [84], where $\boldsymbol{I}$ is an identity matrix. It samples a subset $\boldsymbol{y} \subseteq \mathcal{Y}$ with probability

$$P_{\boldsymbol{L}}(Y = \boldsymbol{y}; \boldsymbol{L}) = \frac{\det(\boldsymbol{L}_{\boldsymbol{y}})}{\det(\boldsymbol{L} + \boldsymbol{I})}, \tag{2.2}$$

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

A vanilla DPP gave rise to state-of-the-art performance on document summarization [85, 86]. Its variation, Markov DPP [87], was used to maintain the diversity between multiple draws from the ground set. A sequential DPP (seqDPP) [1] was proposed for video summarization. Our SH-DPP brings a hierarchy to seqDPP and uses the first layer to take account of the user queries in the summarization (subset selection) process.

2.6.1 Sequential DPP (SeqDPP)

Gong et al. proposed SeqDPP [88] 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 $\boldsymbol{x}_{t-1}\subseteq\mathcal{V}_{t-1}$ selected from the previous partition. In particular, the distribution of the subset selection variable X_t is given by a conditional DPP,

$$\begin{aligned}P(X_t=\boldsymbol{x}_t|X_{t-1}=\boldsymbol{x}_{t-1}) :=&P_L(Y_t=\boldsymbol{x}_t\cup\boldsymbol{x}_{t-1}|\boldsymbol{x}_{t-1}\subseteq Y_t;\boldsymbol{L}^t)\\ =&P_L(X_t=\boldsymbol{x}_t;\boldsymbol{\Omega}^t)=\frac{\det\boldsymbol{\Omega}^t_{\boldsymbol{x}_t}}{\det(\boldsymbol{\Omega}^t+\boldsymbol{I})},\end{aligned} \tag{2.3}$$

where $P_L(Y_t;\boldsymbol{L}^t)$ and $P_L(X_t;\boldsymbol{\Omega}^t)$ are two L-ensemble DPPs with the ground sets $\boldsymbol{x}_{t-1}\cup\mathcal{V}_t$ 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 $\boldsymbol{L}^t$ and $\boldsymbol{\Omega}^t$ is given by [84],

$$\boldsymbol{\Omega}^t=\left([(\boldsymbol{L}^t+\boldsymbol{I}_{\mathcal{V}_t})^{-1}]_{\mathcal{V}_t}\right)^{-1}-\boldsymbol{I}, \tag{2.4}$$

where $\boldsymbol{I}_{\mathcal{V}_t}$ is an identity matrix of the same size as $\boldsymbol{L}^t$ except that the diagonal entries corresponding to $\boldsymbol{x}_{t-1}$ 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 $\boldsymbol{I}$ equals the size of the t-th video segment $\mathcal{V}_t$.

2.7 Summary

This Chapter began with a detailed literature review to the problem of video summarization in Section 2.1. We concluded this Chapter by offering a detailed review of determinantal point process and its extension in Section 2.6.

CHAPTER 3: QUERY-FOCUSED VIDEO SUMMARIZATION

The results of this Chapter have been published in the following paper:

Aidean Sharghi, Boqing Gong, Mubarak Shah, *"Query-Focused Extractive Video Summarization"* in European Conference on Computer Vision (ECCV), 2016, pp. 3–19.[38]

In respect to the recent progress, the goal of this Chapter is to further advance the user-oriented video summarization by modeling user input, or more precisely user intentions, in the summarization process. To this end, we develop a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP), for query-focused extractive video summarization. It is developed upon seqDPP in order to take advantage of seqDPP's properties, and yet rectifies its downside by a two-layer hierarchy. Given a user query and a long video sequence, our algorithm returns a summary by selecting key shots from the video. The decision to include a shot in the summary depends on the shot's relevance to the user query and importance in the context of the video, jointly. We verify our approach on two densely annotated video datasets. The query-focused video summarization is particularly useful for search engines, e.g., to display snippets of videos.

3.1 Methodology

Our approach takes as input a user query q (i.e., concepts) and a long video $\mathcal{Y}$, and outputs a query-focused short summary $y(q, \mathcal{Y})$,

$$y(q, \mathcal{Y}) \leftarrow \operatorname*{argmax}_{y \subseteq \mathcal{Y}} \; P(Y = y | q, \mathcal{Y}), \tag{3.1}$$

which consists of some shots of the video. We desire **four major properties** from the distribution $P(Y = y|q, \mathcal{Y})$. i) It models the subset selection variable Y. ii) It promotes diversity among the items selected by Y. iii) It works efficiently given very long (e.g., egocentric) or even endlessly streaming (e.g., surveillance) videos. iv) It has some mechanism for accepting the user input q. Together, the properties motivate a Sequential and Hierarchical DPP (SH-DPP) as our implementation to $P(Y = y|q, \mathcal{Y})$. SH-DPP is built upon SeqDPP [1]. Therefore, we firstly discuss some related methods—especially SeqDPP, how they meet some of the **properties** but not all, and then present the details of SH-DPP.

3.1.1 Sequential DPP (SeqDPP) with User Queries

In order to satisfy the first two **properties** i) and ii), one can use a vanilla DPP (cf. eq. (2.2)) to extract a diverse subset of shots as a video summary. Though this works well for multi-document summarization [85], it is unappealing in our context mainly due to two reasons. First, DPP sees the ground set (i.e., all shots in a video) as a bag, in which the permutation of the items has no effect on the output. In other words, the temporal flow of the video is totally ignored by DPP; it returns the same summary even if the shots are randomly shuffled. Second, the inference (eq. (3.1)) cost is extremely high when the video is very long, no matter by exhaustive search among all possible subsets $y \subseteq \mathcal{Y}$ or greedy search [30]. We note that the submodular functions also suffer from the same drawbacks [28, 23].

The SeqDPP method [1] meets **properties** i)–iii) and solves the problems described above. It partitions a video into T consecutive disjoint sets, $\cup_{t=1}^{T}\mathcal{Y}_t = \mathcal{Y}$, where $\mathcal{Y}_t$ represents a set consisting of only a few shots and stands as the ground set of time step t. The model is defined as follows (see the left panel of Figure 3.1 for the graphical model),

$$P_{\text{SEQ}}(Y|\mathcal{Y}) = P(Y_1|\mathcal{Y}_1)\prod_{t=2}^{T} P(Y_t|Y_{t-1}, \mathcal{Y}_t), \quad \mathcal{Y} = \cup_{t=1}^{T}\mathcal{Y}_t \tag{3.2}$$

where $P(Y_t|Y_{t-1}, \mathcal{Y}_t) \propto \det \boldsymbol{\Omega}_{Y_{t-1}\cup Y_t}$ is a conditional DPP to ensure diversity between the items selected at time step t (by Y_t) and those of the previous time step (by Y_{t-1}). Similar to the vanilla DPP (cf. eq. (2.2)), here the conditional DPP is also associated with a kernel matrix $\boldsymbol{\Omega}$. In [1], this matrix is parameterized by $\boldsymbol{\Omega}_{ij} = \boldsymbol{f}_i^T W^T W \boldsymbol{f}_j$, where $\boldsymbol{f}_i$ is a feature vector of the i-th video shot and W is learned from the user summaries. Note that the SeqDPP summarizer $P_{\text{SEQ}}(Y|\mathcal{Y})$ does not account for any user input. It is learned from "oracle" summaries in the hope of reaching a good compromise between distinct users.

Here, we instead aim to infer individual users' intentional preferences over the video summaries, through the information conveyed by the user queries. To this end, a simple extension to SeqDPP is to engineer query-dependent feature vectors $\boldsymbol{f}(q)$ of the video shots—details are postponed to Section 3.2.4. We consider this SeqDPP variation as our baseline. It is indeed responsive to the queries through the query-dependent features, but it is fundamentally limited in modeling the query-relevant summaries, in which the importance of a video shot is jointly determined by its relevance to the query and how it is representative in the context of the video. The SeqDPP offers no explicit treatment to the two types of interplayed strengths. In addition, the user may likely expect different levels of diversity from the relevant shots and irrelevant ones, respectively. However, a single DPP kernel in SeqDPP fails to offer such flexibility.

Our SH-DPP possesses all of the four **properties**. It is developed upon SeqDPP in order to take advantage of SeqDPP's nice properties i)–iii), and yet rectifies its downside (mainly on **property** iv)) by a two-layer hierarchy.

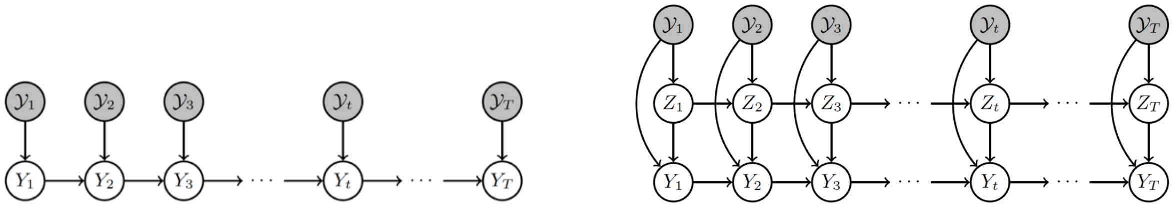

$\mathcal{Y}_1$ $\mathcal{Y}_2$ $\mathcal{Y}_3$ $\mathcal{Y}_t$ $\mathcal{Y}_T$
Y_1 Y_2 Y_3 ··· Y_t ··· Y_T
$\mathcal{Y}_1$ $\mathcal{Y}_2$ $\mathcal{Y}_3$ $\mathcal{Y}_t$ $\mathcal{Y}_T$
Z_1 Z_2 Z_3 ··· Z_t ··· Z_T
Y_1 Y_2 Y_3 ··· Y_t ··· Y_T

Figure 3.1: The graphical models of SeqDPP [1] (left) and our SH-DPP (right).

3.1.2 Sequential and Hierarchical DPP (SH-DPP)

The right panel of Figure 3.1 depicts the graphical model of SH-DPP, reading as,

$$\begin{aligned}&P_{\text{SH}}(\{Y_1, Z_1\}, ..., \{Y_T, Z_T\}|q, \mathcal{Y})\\=&P(Z_1|q, \mathcal{Y}_1)P(Y_1|Z_1, \mathcal{Y}_1)\prod_{t=2}^{T} P(Z_t|q, Z_{t-1}, \mathcal{Y}_t)P(Y_t|Z_t, Y_{t-1}, \mathcal{Y}_t).\end{aligned} \tag{3.3}$$

Query q is omitted from Figure 3.1 for clarity. Each shaded node represents a video segment $\mathcal{Y}_t$ (i.e., a few consecutive and disjoint shots). We first use the subset selection variables Z_t to select the query-relevant video shots. Note that Z_t will choose zero shot if the segment $\mathcal{Y}_t$ does not contain any visual content related to the query. Depending on the results of Z_t (and Y_{t-1}), the variable Y_t at the last layer selects video shots to further summarize the remaining content in the video segment $\mathcal{Y}_t$. The arrows in each layer impose diversity by DPP between the shots selected from two adjacent video segments—we thus have Markov diversities, in contrast to global diversity, in order to allow two (or more) similar shots to be simultaneously sampled to the summary if they appear at far-apart time steps (e.g., a man left home in the morning and returned home in the afternoon).

We define two types of DPPs for the two layers of SH-DPP, respectively.

3.1.2.1 Z-Layer: Summarize Query-Relevant Shots

Similar to SeqDPP, we apply a conditional DPP $P(Z_t|q, Z_{t-1}, \mathcal{Y}_t)$ at each time step t over the ground set $\mathcal{Y}_t \cup \{Z_{t-1} = z_{t-1}\}$, where $\mathcal{Y}_t$ consists of all the shots in partition t and z_{t-1} are the shots selected by Z_{t-1}. In other words, the DPP here is conditioned on the selected items z_{t-1} of the previous time step, enforcing diversity between two consecutive time steps,

$$P(Z_t = z_t|q, Z_{t-1} = z_{t-1}, \mathcal{Y}_t) = \frac{\det \mathbf{\Omega}_{z_{t-1} \cup z_t}}{\det(\mathbf{\Omega}_{z_{t-1} \cup \mathcal{Y}_t} + I_t)} \tag{3.4}$$

where I_t is the same as an identity matrix except that its diagonal values are zeros at the entries indexed by z_{t-1}.

Different from SeqDPP, we dedicate the Z-layer to query-relevant shots only. This is achieved by how we train SH-DPP (Section 3.1.3) and the way we parameterize the DPP kernel matrix,

$$\mathbf{\Omega}_{ij} = [\boldsymbol{f}_i(q)]^T W^T W[\boldsymbol{f}_j(q)] \tag{3.5}$$

where $\boldsymbol{f}(q)$ is a query-dependent feature vector of a shot. Details about the features are postponed to Section 3.2.4. In testing, the Z-layer only selects shots that are relevant to the user query q, and leaves all the unselected shots to the Y-layer which operates like a supplementary summarizer to the Z-layer.

3.1.2.2 Y-Layer: Summarizing the Remaining Video

The decision to include a shot in the query-focused video summarization is driven by two interplayed forces. One is the shot's relevance to the query and the other is the representativeness of the shot. Given a user query q (e.g., CAR+FLOWER) and a long video $\mathcal{Y}$, likely many video shots are irrelevant to the query. As a result, we need another Y-layer to compensate the query-relevant shots selected by the Z-layer. In particular, we define the conditional probability distribution for the Y-layer variables as,

$$P(Y_t = y_t | Y_{t-1} = y_{t-1}, Z_t = z_t, \mathcal{Y}_t) = \frac{\det \boldsymbol{\Upsilon}_{y_{t-1} \cup z_t \cup y_t}}{\det(\boldsymbol{\Upsilon}_{y_{t-1} \cup \mathcal{Y}_t} + I'_t)} \tag{3.6}$$

where y_{t-1} is the selected subset in previous time step at the Y-layer, z_t is the selected subset of query-relevant shots in current time step by the Z-layer, and I'_t is a diagonal matrix with ones indexed by $\mathcal{Y}_t \setminus z_t$ and zeros everywhere else.

Conditioning the Y-layer on the Z-layer has two advantages. First, no redundant information that is already selected by Z-layer is added by the Y-layer again to the summary, i.e., the shots selected by Y-layer are diverse from those by Z-layer. Second, Y-layer can, to some extent, compensate the missed query-relevant shots by Z-layer that were supposed to be selected. This occurs when such shots are themselves important in the context of the video.

Note that the Y-layer involves a new DPP kernel $\boldsymbol{\Upsilon}$, different from that used for the Z-layer. One obvious reason for doing this is that the two layers of variables serve to select different (query relevant or important) types of shots. Besides, it is also worth mentioning that the user may expect various levels of diversity from summary. When a user searches for CAR+FLOWER, s/he probably would like to see some sort of redundancy in the shots of wedding car but not in the shots of police, making it necessary to have two types of DPP kernels.

The Y-layer kernel is parameterized by

$$\boldsymbol{\Upsilon}_{ij} = \boldsymbol{f}_i^T \boldsymbol{V}^T \boldsymbol{V} \boldsymbol{f}_j \tag{3.7}$$

and we will discuss how to extract features $\boldsymbol{f}$ from a shot in Section 3.2.4.

3.1.3 Training and Testing SH-DPP

The training data in our experiments are in the form of $(q, \mathcal{Y}, z^q, y^q)$, where z^q and y^q respectively denote the query relevant and irrelevant shots in the summary. We learn the model parameters $\boldsymbol{W}$ and $\boldsymbol{V}$ of SH-DPP by maximum likelihood estimation (MLE):

$$\max_{\boldsymbol{W},\boldsymbol{V}} \quad \sum_{q}\sum_{\mathcal{Y}} \log P_{\text{SH}}(\{y_1, z_1\}, \cdots, \{y_T, z_T\}|q, \mathcal{Y}) - \lambda_1\|\boldsymbol{W}\|_F^2 - \lambda_2\|\boldsymbol{V}\|_F^2, \tag{3.8}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm. We tune the hyper-parameters λ_1 and λ_2 by a leaving-one-video-out strategy. We use gradient descent to maximize the log likelihood of Equation 3.8. We define:

$$\begin{aligned}&\mathcal{J}^t(\boldsymbol{\Theta}; \mathcal{Y}_t, y_t, y_{t-1}, z_t, z_{t-1}) \equiv \log\ P(Z_t|q, Z_{t-1}, \mathcal{Y}_t) + \log\ P(Y_t|Z_t, Y_{t-1}, \mathcal{Y}_t) = \\ &\log\ \det(\boldsymbol{\Omega}_{z_t\cup z_{t-1}}) - \log\ \det(\boldsymbol{\Omega}_{z_{t-1}\cup\mathcal{Y}_t} + I_t) + \log\ \det(\boldsymbol{\Upsilon}_{z_t\cup y_{t-1}\cup y_t}) - \log\ \det(\boldsymbol{\Upsilon}_{y_{t-1}\cup\mathcal{Y}_t} + I'_t)\end{aligned} \tag{3.9}$$

Now we can write:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\Theta}} = \sum_{t=1}^{T} \frac{\partial\mathcal{J}^t}{\partial\boldsymbol{\Omega}_t}\frac{\partial\boldsymbol{\Omega}_t}{\partial\boldsymbol{\Theta}} + \frac{\partial\mathcal{J}^t}{\partial\boldsymbol{\Upsilon}_t}\frac{\partial\boldsymbol{\Upsilon}_t}{\partial\boldsymbol{\Theta}} = \sum_{t=1}^{T}\sum_{ij} \frac{\partial\mathcal{J}^t}{\partial\boldsymbol{\Omega}_{t_{ij}}}\frac{\partial\boldsymbol{\Omega}_{t_{ij}}}{\partial\boldsymbol{\Theta}} + \sum_{ij} \frac{\partial\mathcal{J}^t}{\partial\boldsymbol{\Upsilon}_{t_{ij}}}\frac{\partial\boldsymbol{\Upsilon}_{t_{ij}}}{\partial\boldsymbol{\Theta}} \tag{3.10}$$

where $\boldsymbol{\Omega}_t$ and $\boldsymbol{\Upsilon}_t$ are the shortened versions of $\boldsymbol{\Omega}_{z_{t-1}\cup\mathcal{Y}_t}$ and $\boldsymbol{\Upsilon}_{y_{t-1}\cup\mathcal{Y}_t}$, respectively. Note that the chain rule has been applied to decompose the gradient into two components, $\boldsymbol{\Omega}_t$ and $\boldsymbol{\Upsilon}_t$. Now for each component, we can write:

$$\frac{\partial\mathcal{J}^t}{\partial\boldsymbol{\Omega}_t} = \frac{\partial\log\ \det(\boldsymbol{\Omega}_{z_t\cup z_{t-1}})}{\partial\boldsymbol{\Omega}_t} - \frac{\partial\log\ \det(\boldsymbol{\Omega}_t + I_t)}{\partial\boldsymbol{\Omega}_t} = \mathcal{M}((\boldsymbol{\Omega}_{z_t\cup z_{t-1}})^{-1}) - (\boldsymbol{\Omega}_t + I_t)^{-1} \equiv \boldsymbol{J_1} \tag{3.11}$$

$$\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{\Upsilon}_t} = \frac{\partial \log\det(\boldsymbol{\Upsilon}_{z_t \cup y_{t-1} \cup y_t})}{\partial \boldsymbol{\Upsilon}_t} - \frac{\partial \log\det(\boldsymbol{\Upsilon}_t + I'_t)}{\partial \boldsymbol{\Upsilon}_t} = \mathcal{M}((\boldsymbol{\Upsilon}_{z_t \cup y_{t-1} \cup y_t})^{-1}) - (\boldsymbol{\Upsilon}_t + I'_t)^{-1}$$
$$\equiv \boldsymbol{J_2} \tag{3.12}$$

where $\mathcal{M}$ is an operator that maps square submatrix A_y to the matrix B such that 1)B has the same dimension as A and 2)$A_y = B_y$ while other elements of B are zero. Recall that we used a linear transformation for parameterizing our DPP kernels:

$$\boldsymbol{\Omega}_{t_{ij}} = \boldsymbol{f}_i(q)^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{f}_j(q) \quad , \quad \boldsymbol{\Upsilon}_{t_{ij}} = \boldsymbol{f}_i^T \boldsymbol{V}^T \boldsymbol{V} \boldsymbol{f}_j$$

For this parameterization of the kernels, we have:

$$\frac{\partial \boldsymbol{\Omega}_{t_{ij}}}{\partial \mathbf{W}} = \mathbf{W}(\boldsymbol{f}_i(q)\boldsymbol{f}_j(q)^T + \boldsymbol{f}_j(q)\boldsymbol{f}_i(q)^T) \quad , \quad \frac{\partial \boldsymbol{\Upsilon}_{t_{ij}}}{\partial \mathbf{V}} = \mathbf{V}(\boldsymbol{f}_i \boldsymbol{f}_j^T + \boldsymbol{f}_j \boldsymbol{f}_i^T) \tag{3.13}$$

Putting all the pieces together, we derive the gradients as following:

$$\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{\Omega}_t} \frac{\partial \boldsymbol{\Omega}_t}{\partial \mathbf{W}} = \sum_{ij} \frac{\partial \mathcal{J}^t}{\partial \boldsymbol{\Omega}_{t_{ij}}} \frac{\partial \boldsymbol{\Omega}_{t_{ij}}}{\partial \mathbf{W}} = \sum_{ij} \boldsymbol{J}_{1_{ij}} \mathbf{W}(\boldsymbol{f}_i(q)\boldsymbol{f}_j^T(q) + \boldsymbol{f}_i q_j \boldsymbol{f}_{q_i}^T) = 2\mathbf{W}\mathbf{F}(q)\mathbf{J}_1\mathbf{F}^T(q) \tag{3.14}$$

Similarly:

$$\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{\Upsilon}_t} \frac{\partial \boldsymbol{\Upsilon}_t}{\partial \mathbf{V}} = \sum_{ij} \frac{\partial \mathcal{J}^t}{\partial \boldsymbol{\Upsilon}_{t_{ij}}} \frac{\partial \boldsymbol{\Upsilon}_{t_{ij}}}{\partial \mathbf{V}} = \sum_{ij} \boldsymbol{J}_{2_{ij}} \mathbf{V}(\boldsymbol{f}_i \boldsymbol{f}_j^T + \boldsymbol{f}_j \boldsymbol{f}_i^T) = 2\mathbf{V}\mathbf{F}\mathbf{J}_2\mathbf{F}^T \tag{3.15}$$

The resulting optimization problem is non-convex. We thus try different initializations and choose the best one using the same leave-one-video-out validation strategy as we used for selecting hyper-parameters. Specifically, out of 3 training videos, 2 are used to train the model, and the remaining video is used for validation. By doing this three times each for a training video, the initialization that achieves the best overall performance is chosen to re-train the model on all 3 training videos.

We then test the obtained model on the test video.

It is worth mentioning that the gradients of the regularization terms are omitted due to simplicity. We used *minFunc* (`https://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`), a tool designed for unconstrained optimization of differentiable real-valued multivariate functions using line-search methods, to optimize the objective function. A maximum of 1000 function evaluations is allowed and additionally, the optimization is terminated whenever the error/progress is less than $1e^{-12}$.

Algorithm 1 Training SH-DPP

```
Require: $F$ video partition features, $q$ the encoded query
Ensure: $\mathcal{J}^t$, $\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{W}}$, $\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{V}}$
  Obtain $\boldsymbol{\Omega}$ and $\boldsymbol{\Upsilon}$ via Eq (3.5) and Eq (3.7)
  Calculate $\mathcal{J}^t$ via Eq (3.9)
  Calculate $\boldsymbol{J_1}$ and $\boldsymbol{J_2}$ via Eq (3.11) and Eq (3.12)
  Calculate $\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{W}}$ and $\frac{\partial \mathcal{J}^t}{\partial \boldsymbol{V}}$ via Eq (3.14) and Eq (3.15)
  Update parameters
```

After obtaining the local optimum $\boldsymbol{W}^*$ and $\boldsymbol{V}^*$ from the training, we need to know how to maximize the SH-DPP $P_{\text{SH}}(y|q,\mathcal{Y})$ for the testing stage (cf. eq. (3.1)). However, SH-DPP remains a computationally extensive combinatorial problem. We thus follow [1] to have an approximate online inference procedure:

$$\begin{aligned}
{z_1}^* &= \operatorname*{argmax}_{z\in\mathcal{Y}_1} P(Z_1|q,\mathcal{Y}_1), & {y_1}^* &= \operatorname*{argmax}_{y\in\mathcal{Y}_1\backslash z_1^*} P(Y_1|z_1^*,\mathcal{Y}_1) \\
{z_t}^* &= \operatorname*{argmax}_{z\in\mathcal{Y}_t} P(Z_t|q,z_{t-1}^*,\mathcal{Y}_t), & {y_t}^* &= \operatorname*{argmax}_{y\in\mathcal{Y}_t\backslash z_t^*} P(Y_t|z_t^*,y_{t-1}^*,\mathcal{Y}_t), \quad t\geq 2
\end{aligned} \tag{3.16}$$

where we exhaustively search for z_t^* and y_t^* from $\mathcal{Y}_t$ at each time step. Thanks to the online inference, we can readily use SH-DPP to handle very long or even endlessly streaming videos.

3.2 Experimental Setup

In this section, we describe the datasets, features of a video shot, user queries, query-focused video summaries for training/evaluation, and finally, which metrics to evaluate our learned video summarizer SH-DPP.

3.2.1 Datasets

We use the UT Egocentric (UTE) dataset [31] and TV episodes [32] whose dense user annotations are provided in [32]. The UTE dataset includes four daily life egocentric videos, each 3–5 hours long, and the TV episodes contain four videos, each roughly 45 minutes long. These two datasets are very different in nature. The videos in UTE are long and recorded in an uncontrolled environment from the first-person view. As a result, many of the visual scenes are repetitive and likely unwanted in the user summaries. On the other hand, the TV videos are the professional episodes of TV series from the third person's viewpoint. All the scenes are well planned and controlled and mostly concise. A good summarization method should be able to work/learn well in both scenarios.

In [32], all the UTE/TV videos are partitioned to 5/10-second shots, respectively, and for each shot a textual description is provided by a human subject. Additionally, for each video, 3 reference summaries are also provided each as a subset of the textual annotations. Thanks to the dense text annotations for every shot, we are able to derive from the text both user queries and two types of query-focused video summaries, respectively, for patient and impatient users.

Table 3.1: The concepts used in our experiments for the UTE and TV datasets.

	Concepts
UTE	area, band, bathroom, beach, bed, beer, blonde, boat, book, box, building, car card, cars, chair, chest, children, chocolate, comics, cross, cup, desk, drink, eggs face, feet, flowers, food, friends, garden, girl, glass, glasses, grass, hair, hall hands, hat, head, house, kids, lady, legs, lights, market, men, mirror, model mushrooms, ocean, office, park, phone, road, room, school, shoes, sign, sky street, student, sun, toy, toys, tree, trees, wall, water, window, windows (70)
TV	accident, animals, area, bat, beer, book, bugs, building, car, card, chair, child clouds, commercial, cross, curves, dance, desk, drink, driver, evening, fan, father food, girls, guy, hands, hat, head, history, home, house, kids, leaves, men, model mother, office, painting, paintings, parents, phone, places, present, room, scene space, storm, street, team, vehicle, wonder (52 in total)

3.2.2 User Queries

Here, a user query comprises one or more noun concepts (e.g., CAR, FLOWER, KID); more generic queries are left for future research. There are many nouns in the text annotations of the video shots, but are they all useful for users to construct queries? Likely no. Any useful nouns have to be machine-detectable so that the system can "understand" the user queries. To this end, we construct a lexicon of concepts by overlapping all the nouns in the annotations with the nouns in SentiBank [89], which is a large collection of visual concepts and corresponding detectors. This results in a lexicon of 70/52 concepts for the UTE/TV dataset (see Table 3.1). Each pair of concepts is considered as a user query for both training and testing our SH-DPP video summarizer. Besides, at the testing phase, we also examine novel queries—all the triples of concepts.

3.2.3 Query-Focused Video Summaries

For each input query and video, we need to know the "groundtruth" video summary for training and evaluating SH-DPP. We construct such summaries based on the "oracle" summaries introduced in [1].

3.2.3.1 Oracle Summary

As mentioned earlier in Section 3.2.1, there are three human-annotated summaries $y^u, u = 1, 2, 3$ for each video $\mathcal{Y}$. An "oracle" summary y^o has the maximum agreement with all of the three annotated summaries, and can be understood as the summary by an "average" user. Such a summary is found by a greedy algorithm [85]. Initialize $y^o = \emptyset$. In each iteration, the set y^o increases by one video shot i which gives rise to the largest marginal gain $G(i)$,

$$y^o \leftarrow y^o \cup \operatorname*{argmax}_{i \in \mathcal{Y}} G(i), \quad G(i) = \sum_u \text{F-score}(y^o \cup i, y_u) - \text{F-score}(y^o, y_u) \tag{3.17}$$

where the F-score follows [32] and is explained in Section 3.2.5. The algorithm stops when there is no such shot that the gain $G(i)$ is greater than 0. Note that thus far the oracle summary is independent of the user query.

3.2.3.2 Query-focused Video Summary

We consider two types of users. A **patient user** would like to watch all the shots relevant to the query in addition to the summary of the other visual content of the video. For example, all the shots whose textual descriptions have the word CAR should be included in the summary if CAR shows up in the query. We union such shots with the oracle summary to have the query-focused summary for the patient user. On the other extreme, an **impatient user** may only want to check the existence of the relevant shots, in contrast to watching all of them. To conduct experiments for the impatient users, we overlap the concepts in the oracle summary with the concept lexicon (cf. Section 3.2.2), and generate all possible bi-concept queries from the survived concepts. Note that the oracle summaries are thus the gold standards for training video summarizers for the impatient users.

3.2.4 Features

We extract high-level concept-oriented features $\boldsymbol{h}$ and contextual features $\boldsymbol{l}$ for a video shot. For each concept in the lexicon (of size 70/52 for the UTE/TV dataset), we firstly use its corresponding SentiBank detector(s) [89] to obtain the detection scores of the key frames, and then average them within each shot. Some of the concepts each maps to more than one detectors. For instance, there are beautiful SKY, clear SKY, and sunny SKY detectors for the concept SKY. We max-pool their shot-level scores, so there is always one detection score, which is between 0 and 1, for each concept. The resultant high-level concept-oriented feature vector $\boldsymbol{h}$ is 70D/52D for a shot of a UTE/TV video. We ℓ_2 normalize it.

Furthermore, we design some contextual features $\boldsymbol{l}$ for a video shot based on the low-level features that SentiBank uses as input to its classifiers. This set of low-level features includes: color histogram, GIST [90], LBP [91], Bag-of-Words descriptor, and an attribute feature [92]. With these features, we put a temporal window around each frame, and compute the mean-correlation as a contextual feature for the frame. The mean-correlation shows how well the frame is representative of the other frames in the temporal window. By varying the window size from 5 to 15 with step size 2, we obtain a 6D feature vector. Again we average pool them within each shot, followed by ℓ_2 normalization, to have the shot-level contextual feature vector $\boldsymbol{l}$.

The concept-oriented and contextual features are concatenated as the overall shot-level feature vector $\boldsymbol{f} \equiv [\boldsymbol{h}; \boldsymbol{l}]$ for parameterizing the DPP kernel of the Y-layer (eq. (3.7)). The Z-layer kernel calls for query-dependent features $\boldsymbol{f}(q)$ (eq. (3.5)). For this purpose, we scale the concept-oriented features according to the query: $\boldsymbol{f}(q) \equiv \boldsymbol{h} \circ \boldsymbol{\alpha}(q)$, where $\circ$ is the element-wise product between two vectors, and the scaling factors $\boldsymbol{\alpha}(q)$ are 1 for the concepts shown in the query and 0.5 otherwise (see Figure 1.1(a, b) for an example). Though we may employ more sophisticated query-dependent features, the simple features scaled by the query perform well in our experiments. The simplicity also enables us to feed the same features to vanilla and sequential DPPs for fair comparison.

3.2.5 Evaluation

We evaluate a system generated video summary by contrasting it against the "groundtruth" summary. The comparison is based on the dense text annotations [32]. In particular, the video summaries are mapped to text representations and then compared by the ROUGE-SU metric [3]. We report the precision, recall, F-measure returned by ROUGE-SU.

In addition, we also introduce a new metric, called *hitting recall*, to evaluate the system summaries from the query-focused perspective. Given the input query q and long video $\mathcal{Y}$, denote by S^q the shots relevant to the query in the "groundtruth" summary, and S^q_{SYSTEM} the query-relevant shots hit by a video summarizer. The hitting recall is calculated by $\text{HR} = |S^q_{\text{SYSTEM}}|/|S^q|$, where $|\cdot|$ is the cardinality of a set. For our SH-DPP model, we report the hitting recall for both the overall summaries and those by the Z-layer only.

Table 3.2: Results of query-focused video summarization with **bi-concept** queries.

Patient Users	UTE (%)					TV episodes (%)				
	F	Prec.	Recall	HR	HR_Z	F	Prec.	Recall	HR	HR_Z
Sampling	**22.12**	**35.07**	17.11	23.61	n/a	27.99	34.75	24.36	16.00	n/a
Ranking	20.66	24.35	18.38	22.05	n/a	32.19	39.96	32.19	16.61	n/a
SubMod [28]	20.98	31.40	26.99	30.10	n/a	32.19	**41.59**	27.01	21.69	n/a
Quasi [35]	12.45	19.47	13.14	14.95	n/a	31.88	27.49	41.69	19.67	n/a
DPP [85]	15.7	19.22	32.08	30.94	n/a	29.62	35.26	34.00	21.29	n/a
SeqDPP [1]	18.85	20.59	35.83	31.91	n/a	27.96	23.80	35.62	14.08	n/a
SH-DPP (ours)	21.27	17.87	**41.65**	**38.26**	**36.92**	**37.02**	38.41	**36.82**	**23.76**	20.35
Impatient Users	UTE (%)					TV episodes (%)				
	F	P	R	HR	HR_Z	F	P	R	HR	HR_Z
Sampling	25.44	44.16	18	6.48	n/a	33.74	**41.03**	28.8	13.03	n/a
Ranking	17.92	21.86	15.46	4.4	n/a	29.67	37.56	24.72	15.43	n/a
SubMod [28]	**27.10**	**51.79**	18.85	8.05	n/a	29.41	38.51	23.85	8.65	n/a
Quasi [35]	11.52	42.32	7.06	1.63	n/a	25.09	27.25	23.71	17.06	n/a
DPP [85]	14.36	30.9	16.18	12.54	n/a	26.01	28.85	39.15	**18.86**	n/a
SeqDPP [1]	12.93	7.89	43.39	12.68	n/a	23.35	16.60	39.69	12.56	n/a
SH-DPP (ours)	25.56	18.51	**45.21**	**22.91**	11.57	**35.36**	30.94	**42.02**	17.07	17.07

3.2.6 Implementation Details

Here we report some details in our implementation of SH-DPP. Out of the four videos in either UTE or TV, we use three videos for training and the remaining one for testing. Each video is taken for testing once and then the averaged results are reported. In the training phase, there are two hyper-parameters in our approach: λ_1 and λ_2 (cf. eq. (3.8)). We choose their values by the leave-one-video-out strategy (over the training videos only). The lower-dimensions of $\boldsymbol{W}$ and $\boldsymbol{V}$ are both fixed to 10, the same as used in SeqDPP [1]. We put 10 shots in the ground set $\mathcal{Y}_t$ at each time step, and also examine the ground sets of the other sizes in the experiments.

We train our model SH-DPP using bi-concept queries. However, we test it using not only the bi-concept queries but also novel three-concept queries.

Table 3.3: Results of query-focused video summarization with novel **three-concept** queries.

Patient Users	UTE (%)			TV episodes (%)		
	F	HR	HR$_Z$	F	HR	HR$_Z$
DPP	20.7	38.53	n/a	29.84	21.68	n/a
SeqDPP	18.03	30.3	n/a	24.29	14.15	n/a
SH-DPP (ours)	**24.54**	**41.23**	40.43	**36.3**	**24.73**	21.71

Impatient Users	UTE (%)			TV episodes (%)		
	F	HR	HR$_Z$	F	HR	HR$_Z$
DPP	14.77	17.28	n/a	24.71	**18.31**	n/a
SeqDPP	19.4	19.17	n/a	29.31	10.09	n/a
SH-DPP (ours)	**29.59**	**25.82**	15.36	**33.94**	17.39	12.33

3.3 Results

This section presents the comparison results of our approach and some competitive baselines, effect of the ground set size, and finally qualitative results.

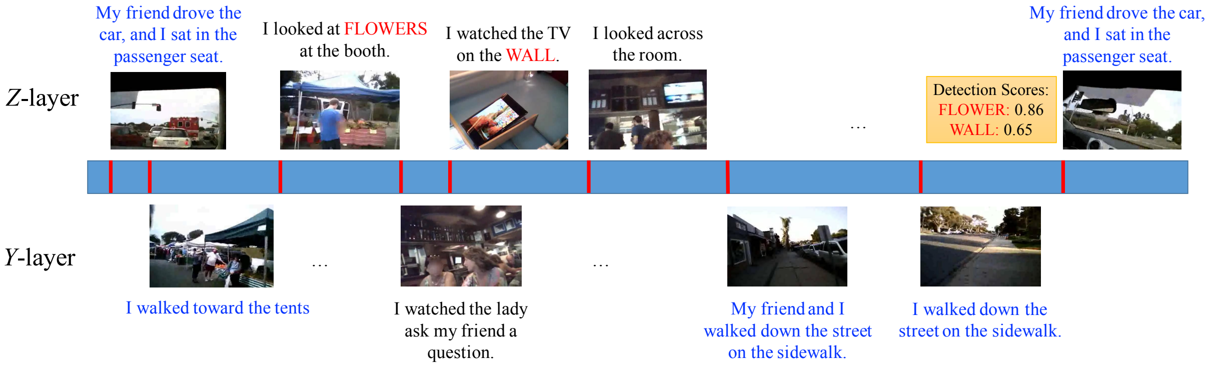
3.3.1 Comparison Results

Table 3.2 shows the results of different summarizers for the query-focused video summarization when the patient and impatient users supply **bi-concept** queries, while Table 3.3 includes the results for novel **three-concept** queries. Note that only bi-concept queries are used to train the summarizers. We report the results on both UTE and TV datasets, and contrast our SH-DPP to the following methods: 1) uniformly sampling K shots, 2) ranking, where for each query we apply the corresponding concept detectors to the shots, assign to a shot a ranking score as the maximum detection score, and then keep the top K shots, 3) vanilla DPP [85], where we remove the dependency between adjacent subset selection variables in Figure 3.1(a), and 4) SeqDPP [1]. We let K be the number of shots in the groundtruth summary; therefore, such privileged information makes 1) and 2) actually strong baselines. We use the same ground sets, whose sizes are fixed to 10 and are studied in Section 3.3.3, for DPP, SeqDPP, and our SH-DPP. All the results are evaluated by the F-measure, Precision, and Recall of ROUGE-SU, as well as the hitting recall (HR) (cf. Section 3.2.5).

Interesting insights can be inferred from Tables 3.2 and 3.3. An immediate observation is that our SH-DPP is able to generate better overall summaries as our average F-measure scores are higher than the others'. Furthermore, our method is able to adapt itself to two essentially different datasets, the UTE daily life egocentric videos and TV episodes.

On UTE, we expect both SH-DPP and SeqDPP to outperform vanilla DPP, because the egocentric videos are very long and include many unwanted scenes, and thus the dependency between different subset selection variables helps eliminate repetitions. In contrast, as mentioned in 3.2.1, the TV episodes are from the world of professional recording, and the scenes rapidly change from shots to shots. Therefore, in this case, the dependency is weak and DPP may be able to catch up SeqDPP's performance. These hypotheses are verified in the results, if we compare DPP with SeqDPP in Tables 3.2 and 3.3.

Another important observation is that in 6 out of the 8 experiments: $\{$patient and impatient users$\}$ on $\{$UTE and TV datasets$\}$ by $\{$bi-concept and novel three-concept queries$\}$, the proposed SH-DPP has better hitting recalls than the other methods, indicating a better response to the user queries. Moreover, the hitting recalls are mainly captured by the Z-layer—the columns HR$_Z$ are the hitting recalls of the shots selected by the Z-layer only of SH-DPP.


Z-layer
My friend drove the car, and I sat in the passenger seat.
I looked at FLOWERS at the booth.
I watched the TV on the WALL.
I looked across the room.
…
Detection Scores:
FLOWER: 0.86
WALL: 0.65
My friend drove the car, and I sat in the passenger seat.
Y-layer
I walked toward the tents
…
I watched the lady ask my friend a question.
…
My friend and I walked down the street on the sidewalk.
I walked down the street on the sidewalk.

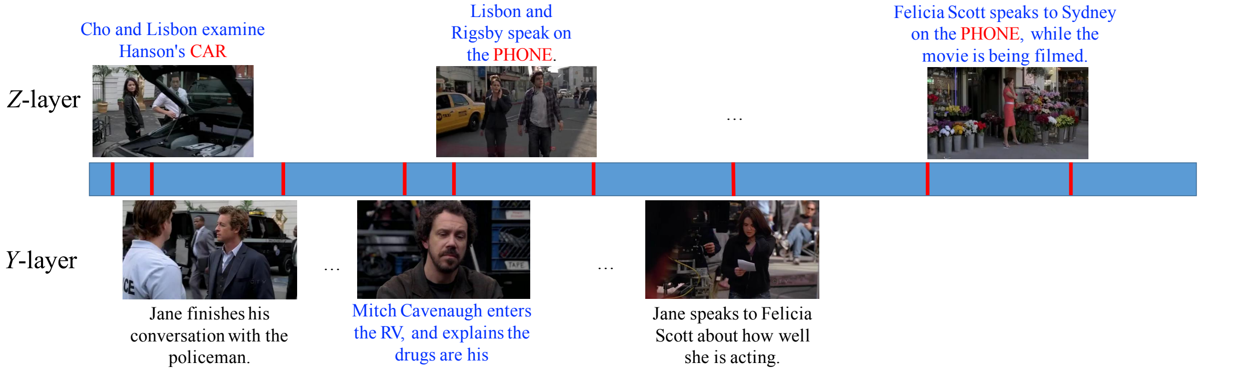
Ground-truth Summary

My friend drove the car, and I sat in the passenger seat. I got out of the car. I walked toward the tents. I looked at the fruit at the booth. My friend and I walked through the market. My friend and I looked at FLOWERS at the booth. My friend drove the car, and I sat in the passenger seat.

I sat with my friend and looked over at the TV on the WALL. I sat at the table while my friend drank. I ate pizza with my friend and we looked at the TV. I looked at the TV on the WALL and then looked back at my friend. I watched the TV on the WALL's at the restaurant.

I walked out the shop with my friend. My friend and I walked down the street on the sidewalk. I walked on the side walk.

Figure 3.2: A peek into SH-DPP. Given the query FLOWERS+WALL, the Z-layer of SH-DPP is supposed to summarize the shots relevant to the query. Conditioning on those results, the Y-layer summarizes the remaining video.


Z-layer
Cho and Lisbon examine Hanson's CAR
Lisbon and Rigsby speak on the PHONE.
…
Felicia Scott speaks to Sydney on the PHONE, while the movie is being filmed.
Y-layer
Jane finishes his conversation with the policeman.
…
Mitch Cavenaugh enters the RV, and explains the drugs are his
…
Jane speaks to Felicia Scott about how well she is acting.

Ground-truth Summary

Jane asks a policeman if Hanson was found with CAR keys or a valet ticket. Jane accuses Yolanda of stealing Hansons valet ticket. Cho and Lisbon examine Hanson's CAR. Jane speaks with Felicia Scott about drugs found at the crime scene. The PHONE rings. Lisbon and Rigsby speak on the PHONE. Mitch Cavenaugh speaks to Jane and Lisbon about the death of Felix Hanson. Mitch Cavenaugh enters the RV, and explains the drugs are his. Felicia Scott speaks to Sydney on the PHONE, while the movie is being filmed. Freddy Ross explains to Cho and Rigsby that Sydney's boyfriend, Brandon got Sydney hooked on drugs. Lisbon questions Sydney about Brandon stealing a gun from Felix Hanson's home. Felicia speaks to Jane about her involvement in Felix Hanson's murder. Felicia Scott speaks to Jane about how she convinced Brandon to kill Felix Hanson. Hanson speaks to Brandon, Brandon shoots Hanson. Felicia Scott speaks to Jane about hurting Sydney.

Figure 3.3: Another peek into a system generated summary by our system. The query includes the concepts CAR and PHONE.

3.3.2 A Peek into the SH-DPP Summarizer

Figure 3.2 is an exemplar summary for the query FLOWERS+WALL by SH-DPP. For each shot in the summary, we show the middle frame of that shot and the corresponding textual description. The groundtruth summary is also included at the bottom half of the figure. We can see that some query-relevant shots are successfully selected by the Z-layer. Conditioning on those, the Y-layer summarizes the remaining video. We highlight the text descriptions (in the blue color) that have exact matches in the groundtruth. However, please note that the other sentences are also highly correlated with some groundtruth sentences, for instance, *"I looked at flowers at the booth"* selected by the Z-layer versus *"my friend and I looked at flowers at the booth"* in the groundtruth summary.

One may wonder why the top-right shot is selected by the Z-layer, since it is visually not relevant to either FLOWERS or WALL. Inspection tells that it is due to the failure of the concept detectors; the detection scores are 0.86 and 0.65 out of 1 for FLOWERS and WALL, respectively. We may improve our SH-DPP for the query-focused video summarization by using better concept detectors.

Figure 3.3 is another peek for the query CAR+PHONE, into SH-DPP, and the corresponding full summary is included in Figure 3.4. We also show another full summary in Figure 3.5 for the query TV+CAR.

Ground-truth summary

Patrick Jane and Teresa Lisbon walk down the street, and enter a crime scene. Rigsby recognizes Hanson, and discusses his past marriage. Jane asks a policeman if Hanson was found with CAR keys or a valet ticket. Jane accuses Yolanda of stealing Hansons valet ticket. Cho and Lisbon examine Hanson's CAR. Rigsby finds drugs in Hanson's CAR. Fanning discusses the previous evening with Lisbon and Jane. Lisbon questions Fanning. Jane explains to Lisbon that Fanning is lying. Lisbon and Van Pelt talk on the PHONE. Jane and Lisbon speak with Felicia Scott about the death of her husband Felix Hanson. Jane speaks with Felicia Scott about drugs found at the crime scene. The PHONE rings. Lisbon and Rigsby speak on the PHONE. Mitch Cavenaugh speaks to Jane and Lisbon about the death of Felix Hanson. Jane and Lisbon speak to Fanning about drugs left out in the RV. Fanning speaks to Jane and Lisbon about his career and drug use. Fanning speaks to Jane and Lisbon about the death of Hanson. Mitch Cavenaugh enters the RV, and explains the drugs are his. Felicia Scott speaks to Sydney on the PHONE, while the movie is being filmed. Cho questions Mitch Cavenaugh about the drugs found in the RV. Cho and Rigsby question Freddy Ross about a PHONE call between him and Hanson. Cho and Rigsby question Freddy Ross about a PHONE call between him and Hanson. Freddy Ross explains to Cho and Rigsby that Sydney's boyfriend, Brandon got Sydney hooked on drugs. Sydney speaks to Lisbon about the drugs in Felix Hanson's CAR found at the murder scene. Lisbon questions Sydney about Brandon stealing a gun from Felix Hanson's home. Jane questions Felicia Scott about Sydney's drug use. Rigsby speaks to Cho while on a stakeout at Felicia Scott's home. Rigsby and Cho chase Brandon as he enters Felicia Scott's home. Cho speaks on the PHONE. Felicia Scott speaks to Rigsby. Sydney sees Brandon laying on the floor. Felicia Scott cries, and Lisbon speaks to Felicia Scott. Cho and Rigsby speak to Lisbon about charging Brandon with Felix Hanson's murder. Jane questions Sydney about Felix Hanson's death and Brandon being shot by Felicia Scott.

System summary

Patrick Jane and Teresa Lisbon meet with Wayne Rigsby and Grace Van Pelt. Lisbon discusses the possibility of the crime being drug related. Jane finishes his conversation with the policeman. Cho and Lisbon examine Hanson's CAR. Jane and Lisbon tell Fanning that Felix Hanson, Hanson, was murdered. Fanning explains that he was celebrating with his friend Felix Hanson. Fanning compliments Hanson's demeanor. Fanning answers Lisbon's questions. Jane explains to Lisbon that Fanning is lying. Jane and Lisbon speak with Felicia Scott about the death of her husband Felix Hanson. Jane questions Felicia Scott about what she liked about Felix Hanson. Jane speaks to Sydney. Lisbon and Rigsby speak on the PHONE. Jane questions Sydney. Cho and Rigsby speak. Freddy Ross drives away. Jane speaks to Mitch Cavenaugh about Felix Hanson. Jane and Lisbon walk away from Mitch Cavenaugh. Jane and Lisbon speak to Fanning about an argument between Fanning and Felix Hanson. Fanning speaks to Jane and Lisbon about his character. Felicia Scott speaks to Sydney on the PHONE, while the movie is being filmed. Sydney drives away from Jane in a golf cart. Cho questions Mitch Cavenaugh about the drugs found in the RV. Mitch Cavenaugh names Freddy Ross as his drug dealer. Cho and Van Pelt look around the room. Rigsby questions Freddy Ross. Freddy Ross speaks to Cho and Rigsby Lisbon questions Sydney. Sydney speaks to Lisbon about the drugs in Felix Hanson's CAR found at the murder scene. Jane questions Sydney about her boyfriend, Brandon. Lisbon questions Felicia Scott about Brandon. Jane questions Felicia Scott about Sydney's drug use. Jane speaks to Lisbon about Felicia Scott. Rigsby speaks to Cho while on a stakeout at Felicia Scott's home. Rigsby and Cho chase Brandon as he enters Felicia Scott's home. Sydney sees Brandon laying on the floor. Cho questions Brandon at the hospital about Felix Hanson. Brandon speaks to Cho. Cho and Rigsby speak to Lisbon about charging Brandon with Felix Hanson's murder. Sydney speaks to Jane. Jane questions Sydney. Felicia Scott speaks to Sydney while filming the movie. Jane speaks to Felicia Scott about how well she is acting.

Figure 3.4: Comparing a groundtruth summary with the corresponding system generated summary. The query includes the concepts CAR and PHONE. The sentences in blue color have exact matches in the summaries, and words marked in red color are the concepts that the user used to compose the query.

Ground-truth summary

My friend drove the CAR, and I sat in the passenger seat. I got out of the CAR. I walked toward the tents. I looked at the fruit at the booth. My friend and I walked through the market. My friend and I sampled a piece of fruit. My friend and I looked at flowers at the booth. My friend drove the CAR, and I sat in the passenger seat. I got out of the CAR. I watched my friend. My friend and I walked down the street on the sidewalk. I looked at the menu with my friend. I sat with my friend and looked over at the TV on the wall. I sat at the table while my friend drank. I ate pizza with my friend and we looked at the TV. I watched the women at the next table. I looked at the TV on the wall and then looked back at my friend. I drank a beer with my friend in a restaurant. I watched the TV on the wall's at the restaurant. My friend and I sat at the table and had food with beer. I walked out the door. I looked at the sidewalk. I walked into the building. My friend and I looked at the frozen yogurt on the scale. My friend and I ate yogurt and watched TV. I ate frozen yogurt. I sat in a chair. I walked out the shop with my friend,. My friend and I walked down the street on the sidewalk. I walked on the side walk. I looked around the street as I walked. I walked down the stairs at the park and looked at a sign. I went for a walk. I walked to the corner. I got into the CAR. I walked to the kitchen. I looked through the instruction booklet. My friend and I played with the LEGO's. I looked through the LEGO's. I played with LEGO's. I opened the book. I stood up. I washed the dishes in the kitchen sink. I washed my hands.

System summary

My friend drove the CAR, and I sat in the passenger seat. I walked toward the tents. I walked toward the parking lot. I walked through the market. I looked at the fruit at the booth. I walked through the market. I looked at some plants. I looked at the flowers at the booth. I looked at the different booths. I took a look around the market. My friend drove the CAR, and I sat in the passenger seat. I looked out the CAR window. My friend and I parked the CAR. I walked around the room. I watched a man place an order. I looked over my friend's shoulder. I looked up at the TV. I looked around the restaurant. I looked across the room. I watched the lady ask my friend a question. I looked up at the pictures. I watched the TV on the wall. I sat at the restaurant. I looked at the front door of the Real Estate office. I tugged on a palm tree leaf. I watched the CARs as they drove by. I looked at the toppings at the dessert shop. I looked out the window. I looked at the TV. I looked at the table. My friend and I stood on the street talking. I walked down the street on the sidewalk. I walked down the stairs at the park and looked at a sign. I looked at all the buildings around me. I looked at the apartment building. I walked through the parking lot. I walked down the street on the sidewalk. I looked at the building. I walked in the parking lot. I looked at the garden. I walked down the street on the sidewalk. I walked on the sidewalk and looked at the street. I looked at the CARs. I looked at the sign. I walked down the street on the sidewalk. My friend and I walked down the street on the sidewalk. My friend drove the CAR, and I sat in the passenger seat. I walked around a dark apartment. I looked up at the ceiling. I forgot to turn on a light. I looked up at the ceiling. I looked at the LEGO's. I looked through the instruction booklet. I read the instructions. I looked at LEGO instructions. I looked at a page in a book and held an object in my hand. I looked at the page of a book. My friend and I played with the LEGO's. I read a book. I sat on the floor and read a book. I washed the dishes in the sink. I fiddled with an object on the counter.

Figure 3.5: Comparing a groundtruth summary with the corresponding system generated summary. The query includes the concepts CAR and TV. The sentences in blue color have exact matches in the summaries, and words marked in red color are the concepts that the user used to compose the query.

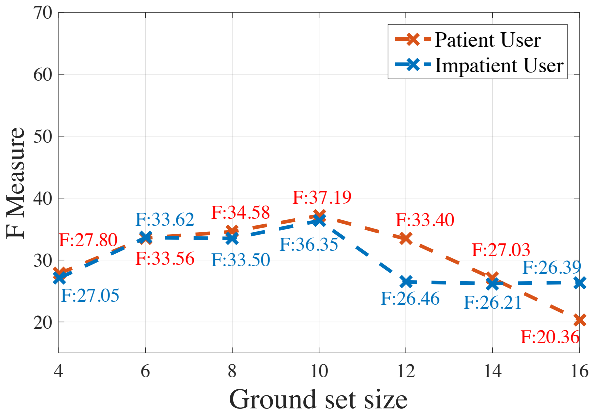


Figure 3.6: The effect of ground set size on the performance of SH-DPP.

3.3.3 Effect of the Ground Set Size

In this section, we examine how changing the ground set size affects the performance of the proposed method. To generate Figure 3.6, we train our SH-DPP with different ground set sizes varying from 4 to 16 by a step size of 2, and then evaluate them on the test data. The results are obtained on the TV episodes where each shot lasts for 10 seconds.

We can see that the F-measure scores only change slightly for the ground sets of 6 to 10 shots. The results decrease a little for very small ground sets (4 shots, 40s) or large sets (16 shots, 160s). This is probably because either too small or too large ground sets fail to follow the internal short memories of the annotators, and further validates our Markov assumption for the video summarization—when users summarize very long videos, they tend to use short memories to impose only local diversities to the selected shots.

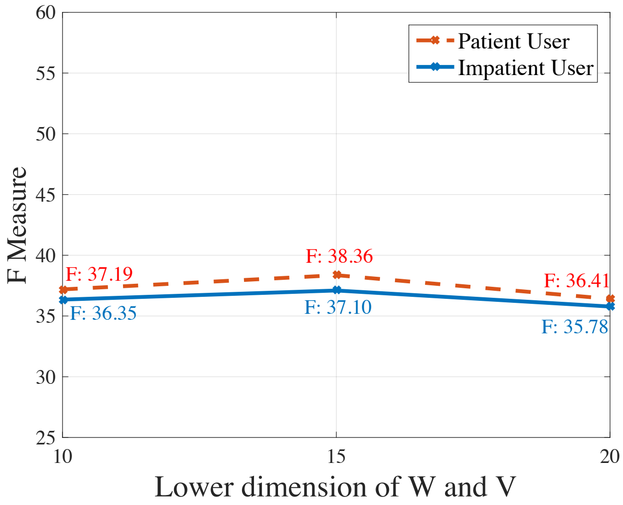


Figure 3.7: Effect of changing the lower dimensions of $\boldsymbol{W}$ and $\boldsymbol{V}$, i.e., the dimension of linearly transformed features. Results are obtained on the TV episodes dataset and evaluated by ROUGE-SU.

3.3.4 Effect of Transformed Feature Dimensions

Figure 3.7 shows the effect of changing the lower dimension of $\boldsymbol{W}$ and $\boldsymbol{V}$, i.e., the dimension of linearly transformed features. We use the same low dimensions for the two matrices. Figure 3.7 shows a good degree of invariance of the results. The results are obtained on the TV episodes dataset and F-measure is used by the ROUGE-SU metric.

3.4 Summary

In this Chapter, we examined a query-focused video summarization problem, in which the decision to select a video shot to the summary depends on both 1) the relevance between the shot and the query and 2) the importance of the shot in the context of the video. To tackle this problem, we developed a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP), as well as efficient learning and inference algorithms for it. Our SH-DPP summarizer can conveniently handle extremely long videos or online streaming videos. On two benchmark datasets for video summarization, our approach significantly outperforms some competing baselines. To the best of our knowledge, ours is the first work on query-focused video summarization, and has a great potential to be used in search engines, e.g., to display snippets of videos.

In the next Chapter, we explore more thoroughly the query-focused video summarization and build a new dataset particularly designed for it. While we collect the user annotations, we meet the challenge how to define a good evaluation metric to contrast system generated summaries to user labeled ones

CHAPTER 4: A MEMORY NETWORK BASED APPROACH FOR QUERY-FOCUSED VIDEO SUMMARIZATION, DATASET, AND EVALUATION

The results of this Chapter have been published in the following paper:

Aidean Sharghi, Jacob S. Laurel, Boqing Gong, *"Query-Focused Video Summarization: Dataset, Evaluation, and A Memory Network Based Approach,"* in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4788-4797.[39]

In previous Chapter, we introduced SH-DPP to perform query-focused video summarization. SH-DPP requires two levels of annotation in user summaries; 1) shots that are generally important, 2) shots that are important because of their relevance to the user query. To remove this costly requirement and to better model the interaction of query with video shots, in this Chapter, we propose a memory network parameterized sequential determinantal point process in order to attend the user query onto different video frames and shots. Unlike our hierarchical model, this approach does not rely on the costly user supervision about which queried concept appears in which video shot or any pre-trained concept detectors.

Second, to better evaluate the performance of video summarizers, we contend that a good evaluation metric for video summarization should focus on the semantic information that humans can perceive rather than the visual features or temporal overlaps. To this end, we collect dense per-video-shot concept annotations, compile a new dataset, and suggest an efficient and automatic evaluation method defined upon the concept annotations. We conduct extensive experiments contrasting our video summarizer to existing ones and present detailed analyses about the dataset and the new evaluation method.

4.1 Dataset

In this section, we provide the details on compiling a comprehensive dataset for video summarization. We opt to build upon the currently existing UT Egocentric (UTE) dataset [33] mainly for two reasons: 1) the videos are consumer grade, captured in uncontrolled everyday scenarios, and 2) each video is 3–5 hours long and contains a diverse set of events, making video summarization a naturally desirable yet challenging task. In what follows, we first explain how we define a dictionary of concepts and determine the best queries over all possibilities for the query-focused video summarization. Then we describe the procedure of gathering user summaries for the queries. We also show informative statistics about the collected dataset.

4.1.1 Concept Dictionary and Queries

We plan to have annotators to transform the semantic information in each video shot to a binary semantic vector (cf. Figures 4.2 and 4.3), with 1's indicating the presence of the corresponding concepts and 0's the absence. Such annotations serve for the sake of an efficient and automatic evaluation method for video summarization (cf. Section 4.1.2.1). The key is thus to have a dictionary that covers a wide range and multiple levels of concepts, in order to have the right basis to encode the semantic information.

Previously, we constructed a lexicon of concepts by overlapping nouns in the video shot captions [2] with those in the SentiBank [89]. Those nouns serve as a great starting point for us since they are mostly entry-level [93] words. We prune out the concepts that are weakly related to visual content (e.g., "AREA", which could be interpreted in various ways and applicable to most situations). Additionally, we merge the redundant concepts such as ”CHILDREN” and ”KIDS”. We also add some new concepts in order to construct an expressive and comprehensive dictionary. Two strategies are employed to find the new concept candidates. First, after watching the videos, we manually add the concepts that appear for a significant amount, e.g., ”COMPUTER”. Second, we use the publicly available statistics about YouTube and Vine search terms to add the terms that are frequently searched by users, e.g., ”PET/ANIMAL”. The final lexicon is a set of 48 concepts (cf. Figure 4.1) that are deemed to be comprehensive for most video genres.

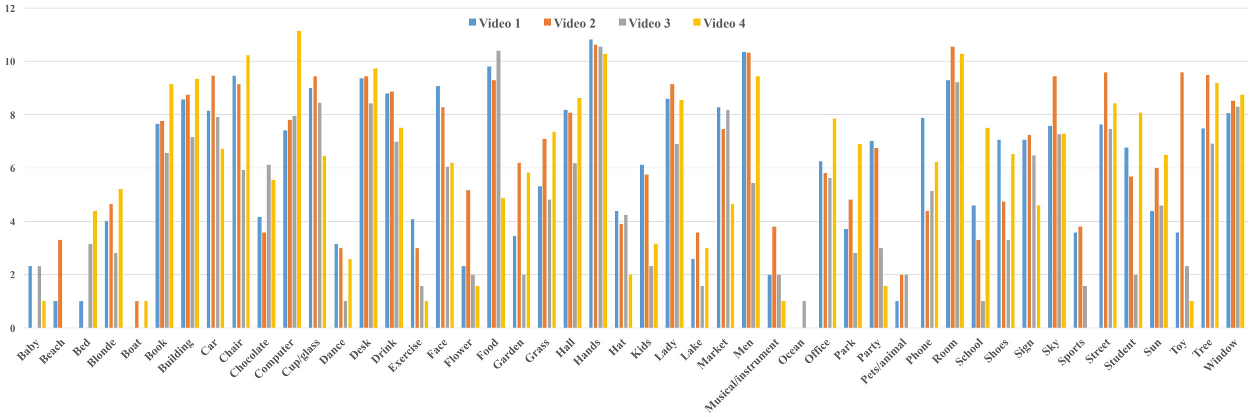


Figure 4.1: The frequencies of concepts showing up in the video shots, counted for each video separately.

We also construct queries, to acquire query-focused user summaries, using two or three concepts as opposed to singletons. Imagine a use case of video search engines. The queries entered by users are often more than one word. For each video, we formalize 46 queries. They cover the following four distinct scenarios: i) all the concepts in the query appear in the same video shots together (15 such queries); ii) all concepts appear in the video but never jointly in a single shot (15 queries), iii) only one of the concepts constituting the query appears in the some shots of the video (15 queries), and iv) none of the concepts in the query are present in the video (1 such query). Such queries and their user annotated summaries thus challenge an artificially intelligent video summarizer from various perspectives.

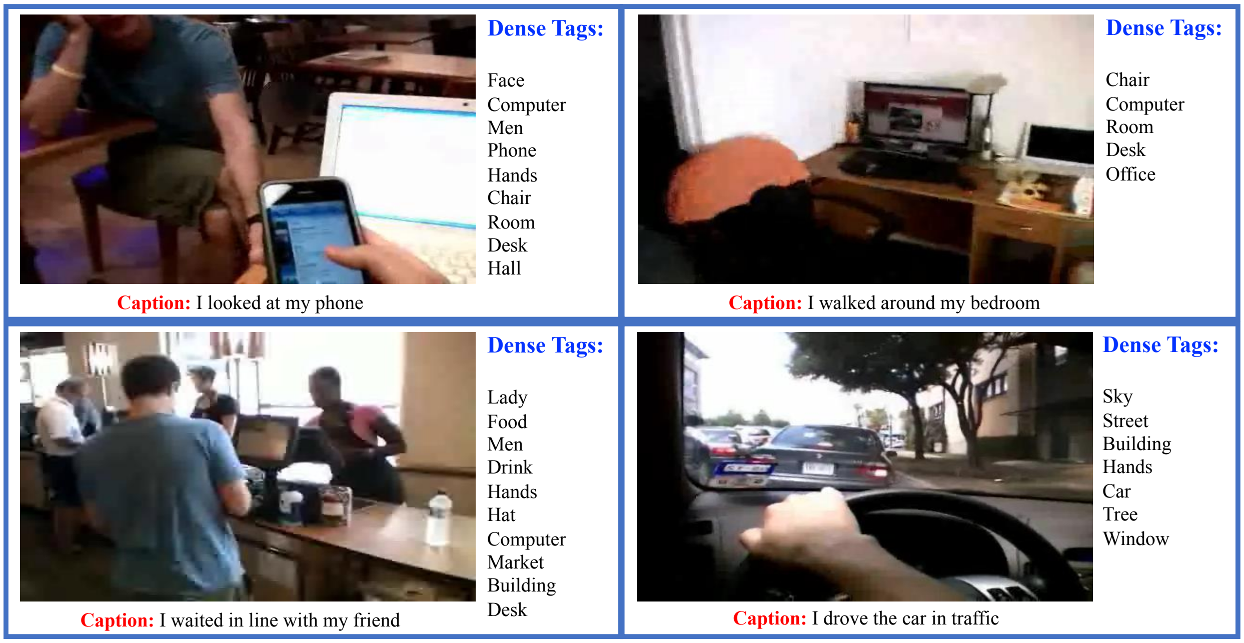


Figure 4.2: Comparing semantic information in our dense tags vs captions provided by [2]. The figure illustrates that the caption is targeting limited information about the scene, while the dense annotations are able to better explain the characteristics of the scene.

By processing the dense user annotation data, we extract various statistics that enable us to have the queries covering a wide range of varieties. Initially, a concept is assumed present in the video if it appears in at least T shots. This is to filter the present noise in annotations acquired from AMT workers and to make sure the concepts really appear together (to steer clear of the pairs that are tagged together as a result of noise or bias).

As described earlier, when a user enters a query q (for instance, on a video search engine), which is usually more than one word, we have four distinct scenarios; i) all the concepts in the query appear in the same video shots together, ii) all concepts appear in the video, but never jointly in a single shot, iii) only one of the concepts constituting the query appears in the some shots of the video, and iv) none of the concepts in the query are present in the video (1 such query). A robust video summarizer, must be able to maintain good performance under any of the scenarios. Therefore, by including enough samples of all the scenarios, we build a comprehensive and diverse dataset.

For the scenario i, we create a list of pairs that appear together in the same shots and sort it in descending order. There are two approaches to select concept pairs from this list: 1) to employ a random selection process where the probability of selecting a pair from the list is proportional to the number of times the pair appeared together in the video (this gives higher chance to the concepts that tend to happen together in the video while not completely crossing out the concepts that are not dominant in the video), and 2) to pick few top concept pairs. We opt to use the random selection process to better generalize the dataset and remove bias.

For the scenario ii, we are interested in concept pairs that are present in the video but not in the same shots, e.g., concept pairs such as CAR and ROOM that are unlikely to appear in the same shots of the video. To this end, for each pair we compute their harmonic mean of frequencies:

$$score(f_{c_1}, f_{c_2}) = \frac{f_{c_1} \times f_{c_2}}{f_{c_1} + f_{c_2}} \tag{4.1}$$

where f_{c_1} and f_{c_2} are the frequencies of concepts c_1 and c_2, respectively. This formulation has two interesting features that make it useful in this regard; 1) the resulting combination of numbers fed to it is always smaller than the smallest entry, 2) it is maximized when both inputs are large and identical. By computing the harmonic mean of frequencies for all the pairs in the list and sorting it in descending order, the concept pairs that have high frequencies for both concepts constituting the query are ranked higher. At this point, we employ the same random selection process to randomly choose pairs from this list.

For the third scenario, we concentrate on pairs that only one concept constituting the query is present in the video, e.g., if there is no CAR present in the entire video while there exists shots with COMPUTER appearing in them, the pair CAR and COMPUTER is a candidate for this scenario. To make sure that the constructed dataset is comprehensive and benefits from the versatile dictionary, we first exclude the concepts that were used in the first two scenarios, we put the rest in a list and use their frequencies to randomly sample from them.

For the last scenario, where neither of concepts in pairs must be present in the video, we simply use the concepts that never appear in the video.

For scenarios i, ii, and iii, we select 15 queries. For scenario iv, we only choose one query; summarizing based on any such query consisting of concepts that are not present in the entire video must result in about the same summary. In other terms, when a user wants the model to summarize the video based on a query consisting of non-present concepts, the summarizer must only return *contextually* important segments of the video, that is essentially what a conventional generic video summarization approach (as opposed to query-dependent approaches) generates.

Figure 4.4 shows that queries play a major role in the summaries that users generate. For a particular video, the same user has selected summaries that have both common (green margin) and uncommon (orange margin) segments.

4.1.2 Collecting User Annotations

We plan to build a video summarization dataset that offers 1) efficient and automatic evaluation metrics and 2) user summaries in response to different queries about the videos. For the former 1), we collect user annotations about the presence/absence of concepts in each video shot. This is a quite daunting task conditioning on the lengths of the videos and the size of our concept dictionary. We use Amazon Mechanical Turk (MTurk) (`http://www.mturk.com/`) for economy and efficiency considerations. For the latter 2), we hire three student volunteers to have better quality control over the labeled video summaries. We partition the videos to 5-second-long shots.

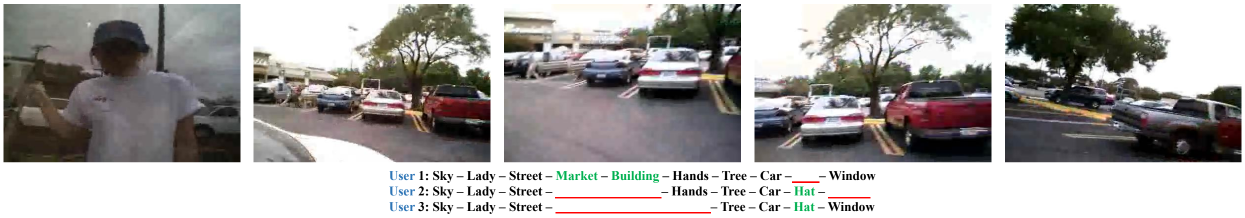

User 1: Sky – Lady – Street – Market – Building – Hands – Tree – Car – ____ – Window
User 2: Sky – Lady – Street – ____________ – Hands – Tree – Car – Hat – ______
User 3: Sky – Lady – Street – ____________________ – Tree – Car – Hat – Window

Figure 4.3: All annotators agree with each other on the prominent concepts in the video shot, while they miss different subtle concepts.

4.1.2.1 Shot Tagging: Visual Content to Semantic Vector

We ask MTurkers to tag each video shot with all the concepts that are present in it. To save the workers' time from watching the shots, we uniformly extract five frames from each shot. A concept is assumed relevant to the shot as long as it is found in any of the five frames. Figure 4.3 illustrates the tagging results for the same shot by three different workers. While all the workers captured the prominent concepts like SKY, LADY, STREET, TREE, and CAR, they missed different subtle ones. The union of all their annotations, however, provides a more comprehensive semantic description about the video shot than that of any individual annotator. Hence, we ask three workers to annotate each shot and take their union to obtain the final semantic vector for the shot. On average, we have acquired 4.13, 3.95, 3.18, and 3.62 concepts per shot for the four videos, respectively. In sharp contrast, the automatically derived concepts [38] from the shot captions [2] are far from enough; on average, there are only 0.29, 0.58, 0.23, and 0.26 concepts respectively associated with each shot of the four videos.

4.1.2.2 Evaluating Video Summaries

Thanks to the dense concept annotations per video shot, we can conveniently contrast a system generated video summary to user summaries according to the semantic information they entail.

We first define a similarity function between any two video shots by intersection-over-union (IOU) of their corresponding concepts. For instance, if one shot is tagged by $\{$CAR, STREET$\}$ and another by $\{$STREET, TREE, SIGN$\}$, then the IOU similarity between them is $1/4 = 0.25$.

To find the match between two summaries, it is convenient to execute it by the maximum weight matching of a bipartite graph, where the summaries are on opposite sides of the graph. The number of matched pairs thus enables us to compute precision, recall, and F1 score. Although this procedure has been used in the previous work [15, 94], there the edge weights are calculated by low-level visual features which by no means match the semantic information humans obtain from the videos. In sharp contrast, we use the IOU similarities defined directly over the user annotated semantic vectors as the edge weights.

4.1.2.3 Acquiring User Summaries

In addition to the dense per-video-shot concept tagging, we also ask annotators to label query-focused video summaries for the 46 queries described in Section 4.1.1.

To ensure consistency in the summaries and better quality control over the summarization process, we switch from MTurk to three student volunteers in our university. We meet and train the volunteers in person. They each summarize all four videos by taking queries into account — an annotator receives 4 (videos) $\times$ 46 (queries) summarization tasks in total. We thus obtain three user summaries for each query-video pair.

However, we acknowledge that it is infeasible to have the annotators to summarize all the query-video pairs from scratch — a video sequence is 3–5 hours long. To overcome this issue, we expand each temporal video to a set of static key frames. First, we uniformly extract five key frames to represent each shot (the same for Section 4.1.2.1). Second, we pool all the shots mentioned in the three textual summaries [2] as the initial candidate set. Third, for each query, we further include all the shots that are relevant to it into the set. As a result, we have a set of candidate shots for each query that covers the main story in the video as well as those of relevant to the query. The annotators summarize the video by removing redundant shots from the set. There are 2500 to 3600 shots in the candidate sets, and the summaries labeled by the volunteers contain only 71 shots on average.

Table 4.1: Inter-user agreement evaluated by F1 score (%) (U1, U2, and U3: the three student volunteers, O: the oracle summary).

U1-U2	U1-U3	U2-U3	U1-O	U2-O	U3-O	
Vid1	58.4	52.7	61.0	66.2	81.7	75.1
Vid2	54.9	62.0	60.3	69.2	75.5	82.3
Vid3	59.1	64.9	69.9	69.7	81.4	86.3
Vid4	48.7	43.8	59.5	54.8	80.4	76.6
Avg	55.27	55.85	62.67	64.97	79.75	80.07

Table 4.2: The average lengths and standard deviations of the summaries for different queries.

	User 1	User 2	User 3	Oracle
Vid1	143.7±32.5	80.2±47.1	62.6±15.7	82.5±33.9
Vid2	103.0±45.0	49.9±25.2	64.4±11.7	64.1±11.7
Vid3	97.3±38.9	50.1±9.6	58.4±9.3	59.2±9.6
Vid4	79.9±30.3	34.4±7.3	28.9±8.7	35.6±8.5

4.1.3 Oracle Summaries

Supervised video summarization methods [1, 28, 38, 34, 36] often learn from one summary per video, or per query-video pair in query-focused summarization, while we have three user generated summaries per query. We aggregate them into one, called the oracle summary, per query-video pair by a greedy algorithm. The algorithm starts from the common shots in the three user summaries. It then greedily chooses one shot every time such that this shot gives rise to the largest marginal gain over the evaluated F1 score. The oracle summaries achieve better agreements with different user summaries (cf. Table 4.1).

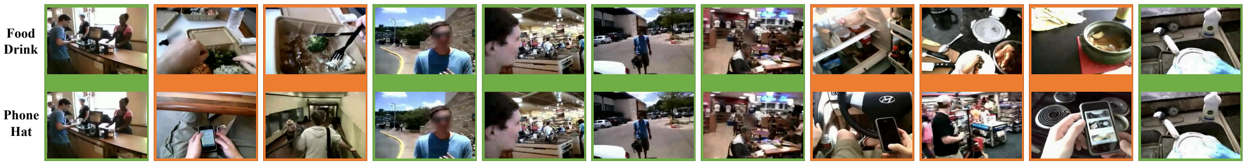


Figure 4.4: Two summaries generated by the same user for two different queries {HAT, PHONE} and {FOOD, DRINK}, respectively. The shots in the two summaries beside the green bars exactly match each others, while the orange bars show the query-specific shots.

4.1.4 Summaries of the Same Video Differ Due to Queries

Figure 4.4 shows two summaries labeled by the same user for two distinct queries, {HAT, PHONE} and {FOOD, DRINK}. Note that the summaries both track the main events happening in the video while they differ in the query-specific parts. Besides, table 4.2 reports the means and standard deviations of the lengths of the summaries per video per user. We can see that the queries highly influence the resulting summaries; the large standard deviations attribute to the queries.

4.1.5 Budgeted Summary

For all the summaries thus far, we do not impose any constraints over the total number of shots to be included into the summaries. After we receive the annotations, however, we let the same volunteers further reduce the lengths of their summaries to respectively 20 shots and 10 shots. We call them *budgeted* summaries and leave them for future research.

4.2 Methodology

We elaborate our approach to the query-focused video summarization in this section. Denote by $\mathcal{V} = \{\mathcal{V}_t\}_{t=1}^{T}$ a video that is partitioned to T segments, and by q the query about the video. In our experiments, every segment $\mathcal{V}_t$ consists of 10 video shots each of which is 5-second long and is used in Section 4.1.2 to collect concept annotations.

4.2.1 Query Conditioned Sequential DPP

The sequential determinantal point process (DPP) [1] is among the state-of-the-art models for generic video summarization. We condition it on the query q as our overarching video summarization model,

$$P(Y_1 = \boldsymbol{y}_1, Y_2 = \boldsymbol{y}_2, \cdots, Y_T = \boldsymbol{y}_T | \mathcal{V}, q) = \quad P(Y_1 = \boldsymbol{y}_1 | \mathcal{V}_1, q) \prod_{t=2}^{T} P(Y_t = \boldsymbol{y}_t | \mathcal{V}_t, \boldsymbol{y}_{t-1}, q) \quad (4.2)$$

where the t-th DPP variable Y_t selects subsets from the t-th segment $\mathcal{V}_t$, i.e., $\boldsymbol{y}_t \subseteq \mathcal{V}_t$, and the distribution $P(Y_t = \boldsymbol{y}_t | \mathcal{V}_t, \boldsymbol{y}_{t-1}, q)$ is specified by a conditional DPP [30],

$$P(Y_t = \boldsymbol{y}_t | \mathcal{V}_t, \boldsymbol{y}_{t-1}, q) = \frac{\det[\boldsymbol{L}(q)]_{\boldsymbol{y}_t \cup \boldsymbol{y}_{t-1}}}{\det\big(\boldsymbol{L}(q) + \boldsymbol{I}_t\big)}. \quad (4.3)$$

The nominator on the right-hand side is the principle minor of the (L-ensemble) kernel matrix $\boldsymbol{L}(q)$ indexed by the subsets $\boldsymbol{y}_t \cup \boldsymbol{y}_{t-1}$. The denominator calculates the determinant of the sum of the kernel matrix and a special identity matrix whose elements indexed by $\boldsymbol{y}_{t-1}$ are 0's. Readers are referred to the great tutorial [30] on DPP for more details.

Note that the DPP kernel $\boldsymbol{L}(q)$ is parameterized by the query q. We have to carefully devise the way of parameterizing it in order to take account of the following properties. In query-focused video summarization, a user selects a shot to the summary for two possible reasons. One is that the shot is quite related to the query and thus becomes appealing to the user. The other may attribute to the contextual importance of the shot; e.g., the user would probably choose a shot to represent a prominent event in the video even if the event is not quite relevant to the query. To this end, we use a memory network to model the two types of importance (query-related and contextual) of a video shot simultaneously.

4.2.2 Memory Network to Parameterize DPP Kernels

The memory network [37] offers a neural network architecture to naturally attend a question to "facts" (cf. the rightmost panel of Figure 4.5). In our work, we shall measure the relevance between the query q and a video shot and incorporate such information into the DPP kernel $\boldsymbol{L}(q)$. Therefore, it is straightforward to substitute the question in memory network by our query, but the "facts" are less obvious.

As discussed in Section 4.1.1, there could be various scenarios for a query and a shot. All the query concepts may appear in the shot but possibly in different frames; one or two concepts of the query may not be present in the shot; it is also possible that none of the concepts are relevant to any frame in the shot. In other words, the memory network is supposed to screen all the video frames in order to determine the shot's relevance to the query. Hence, we uniformly sample 8 frames from each shot as the "facts". The video frames are represented using the same feature as [38] (cf. $\boldsymbol{f}_1, \cdots, \boldsymbol{f}_K$ on the rightmost panel of Figure 4.5).

The memory network takes as input the video frames $\{\boldsymbol{f}_k\}$ of a shot and a query q. The frames are transformed to memory vectors $\{\boldsymbol{m}_k\}$ through an embedding matrix A. Similarly, the query q, represented by a binary indication vector, is mapped to the internal state $\boldsymbol{u}$ using an embedding matrix C. The attention scheme is implemented simply by a dot product followed by a softmax function,

$$p_k = \text{Softmax}(\boldsymbol{u}^T \boldsymbol{m}_k), \tag{4.4}$$

where p_k carries how much attention the query q incurred over the frame $\boldsymbol{f}_k$.

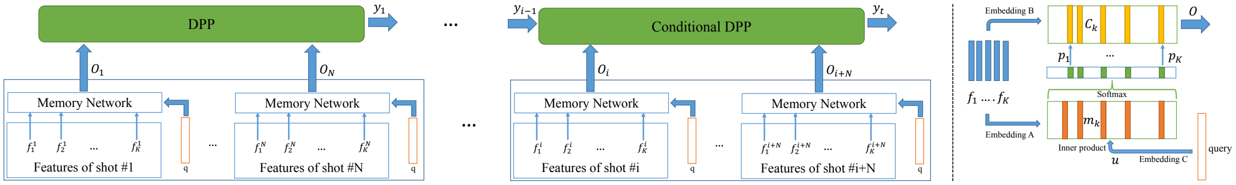


Figure 4.5: Our query-focused video summarizer: Memory network (right) parameterized sequential determinantal point process (left).

Equipped with the attention scores $\{p_k\}$, we assemble another embedding $\{\boldsymbol{c}_k\}$ of the frames, obtained by the mapping matrix B in figure 4.5, into the video shot representation $\boldsymbol{o}$:

$$\boldsymbol{o} = \sum_k p_i \boldsymbol{c}_k, \tag{4.5}$$

which is conditioned on the query q and entails the relevance strength of the shot to the query. As a result, we expect the DPP kernel parameterized by the following

$$[\boldsymbol{L}(q)]_{ij} = \boldsymbol{o}_i^T D^T D \boldsymbol{o}_j \tag{4.6}$$

is also flexible in modeling the importance of the shots to be selected into the video summary. Here i and j index two shots, and D is another embedding matrix. Note that the contextual importance of a shot can be inferred from a shot's similarities to the others by the kernel matrix, while the query-related importance is mainly by the attention scheme in the memory network.

4.2.3 Learning and Inference

We learn the overall video summarizer, including the sequential DPP and the memory network, by maximizing the log-likelihood of the user summaries in the training set. We use stochastic gradient descent with mini-batching to optimize the embedding matrices $\{A, B, C, D\}$. The learning rates and numbers of epochs are chosen using the validation set. At the test stage, we sequentially visit the video segments $\mathcal{V}_1, \cdots, \mathcal{V}_T$ and select shots from them using the learned summarization model.

It is notable that our approach requires less user annotations than the SH-DPP [38]. It learns directly from the user summaries and implicitly attend the queries to the video shots. However, SH-DPP requires very costly annotations about the relevances between video shots and queries. Our new dataset does supply such supervisions, so we shall include SH-DPP as the baseline method in our experiments.

4.3 Experiments

We report experimental setup and results in this section.

4.3.1 Features

We extract the same type of features as used in the existing SH-DPP method [38] in order to have fair comparisons. First, we employ 70 concept detectors from SentiBank [89] and use the detection scores for the features of each key frame (8 key frames per 5-second-long shot). However, it is worth mentioning that our approach is not limited to using concept detection scores and, more importantly unlike SH-DPP, does not rely on the per-shot query annotations to train the summarizer — the per shot user labeled semantic vectors mainly serve for evaluation purpose. Additionally, we extract a six dimensional contextual feature vector per shot by computing the mean-correlations of low-level features (including color histogram, GIST [90], LBP [91], Bag-of-Words, as well as an attribute feature [92]) in a temporal window whose size varies from 5 to 15. The six-dimensional contextual features of a shot are appended to the key frame features of that shot.

4.3.2 Data Split

We run four rounds of experiments each leaving one video out for testing and one for validation, while keeping the remaining two for training. Since our video summarizer and the baselines are sequential models, the small number (i.e., two) of training videos is not an issue as the videos are extremely long, providing many variations and supervisions at the training stage.

4.3.3 Query-Focused Video Summarization

We contrast our video summarizer, the memory-network based sequential determinantal point process, to several closely related methods. We first include SH-DPP [38], the most recent approach to the query-focused video summarization. Our model improves upon SeqDPP [1] by taking the query into account and parameterizing the DPP kernel by the memory network. SeqDPP is thus directly comparable to ours. We concatenate the query features (binary indication vectors) with the shot features and input them to SeqDPP and SH-DPP. We set the same dimensionality for all the embedding spaces in our and the two baseline methods. It turns out the 128D embeddings are chosen due to their performances on the validation videos.

Table 4.3 compares the performances of the three video summarizers. Each video is taken in turn as the test video and the corresponding results are shown in each row. The average results are included as the last row. Precision, recall, and F1 score are reported for all the video summarizers.

Table 4.3: Comparison results for query-focused video summarization (%).

	SeqDPP [1]			SH-DPP [38]			**Ours**		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Vid1	**53.43**	29.81	36.59	50.56	29.64	35.67	49.86	**53.38**	**48.68**
Vid2	**44.05**	46.65	**43.67**	42.13	46.81	42.72	33.71	**62.09**	41.66
Vid3	49.25	17.44	25.26	51.92	29.24	36.51	**55.16**	**62.40**	**56.47**
Vid4	11.14	**63.49**	18.15	11.51	62.88	18.62	**21.39**	63.12	**29.96**
Avg.	39.47	39.35	30.92	39.03	42.14	33.38	**40.03**	**60.25**	**44.19**

Our approach outperforms the other two by a large margin (more than 10% F1 score on average). It seems like Video 4 is especially challenging for all the methods. For Video 2, our summarizer generates a little longer summaries than the others do. In the future work, we will explore how to control the summary length in the sequential DPP model.

4.3.4 Component-wise Analyses

To investigate how each component in our framework contributes to the final results, we conduct more experiments by either removing or modifying them. Figure 4.6 shows the corresponding results.

The main benefit from the memory network is the attention mechanism (cf. equation (4.6)). If we instead use a uniform distribution for the attention scores $\{p_i\}$ and append the query information $\boldsymbol{u}$ directly after the memory network output $\boldsymbol{o}$, the results become worse on all the four videos (cf. NoAttention in Figure 4.6). The NoEmbD results are obtained after we remove the last embedding matrix $\boldsymbol{D}$ when we compute the DPP kernels. Finally, EmbSize 256 are the results when we change the 128D embeddings in our approach to 256D. The performance drops from our complete model verify that all the corresponding components are complementary, jointly contributing to the final results.

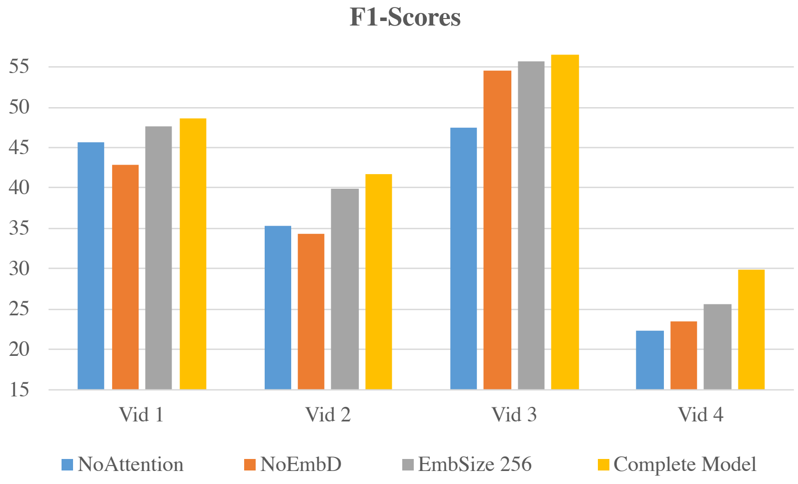


Figure 4.6: The Effectiveness of various individual components in our proposed video summarizer.

4.3.5 Generic Video Summarization

Recall that our queries incur four different scenarios (cf. Section 4.1.1). When there are no video shots relevant to the query, it reduces to the generic video summarization in some sort. We single out such queries and contrast our summarizer to some existing and recent methods for generic video summarization: SubMod [28] which employs submodular functions to encourage diversity and Quasi [35] which is an unsupervised method based on group sparse coding. Unlike the DPP type of summarizers, the baseline methods here are not able to automatically determine the lengths of the summaries. We tune the threshold parameter in Quasi such that the output lengths are no more or less than the oracle summary by 20 shots. For SubMod we set the budget parameter such that it generates summaries that are exactly as long as the oracle summaries. As shown in Table 4.4, our approach still gives the best overall performance even though we reveal the oracle sumamries' lengths to the baseline methods, probably due to its higher neural network based modeling capacity.

Table 4.4: Comparison results for generic video summarization, i.e., when no video shots are relevant to the query.

	SubMod [28]			Quasi [35]			**Ours**		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Vid1	47.86	51.28	49.51	57.37	49.36	53.06	**65.88**	59.75	**62.66**
Vid2	56.53	46.50	51.03	46.75	63.34	**53.80**	35.07	**67.31**	46.11
Vid3	62.46	66.72	64.52	53.93	46.44	49.91	**65.95**	53.12	**58.85**
Vid4	**34.49**	37.25	**35.82**	13.00	**77.88**	22.31	22.29	67.74	33.5
Avg.	**50.34**	50.44	50.22	42.76	59.25	44.77	47.3	**61.98**	**50.29**

4.3.6 A Nice Behavior of Our Evaluation Metric

Our evaluation method for video summarization is mainly motivated by Yeung et al. [2]. Particularly, we share the same opinion that the evaluation should focus on the semantic information which humans can perceive, rather than the low-level visual features or temporal overlaps. However, the captions used in [2] are diverse, making the ROUGE-SU4 evaluation unstable and poorly correlated with human judgments [95], and often miss subtle details (cf. Figure 4.2 for some examples).

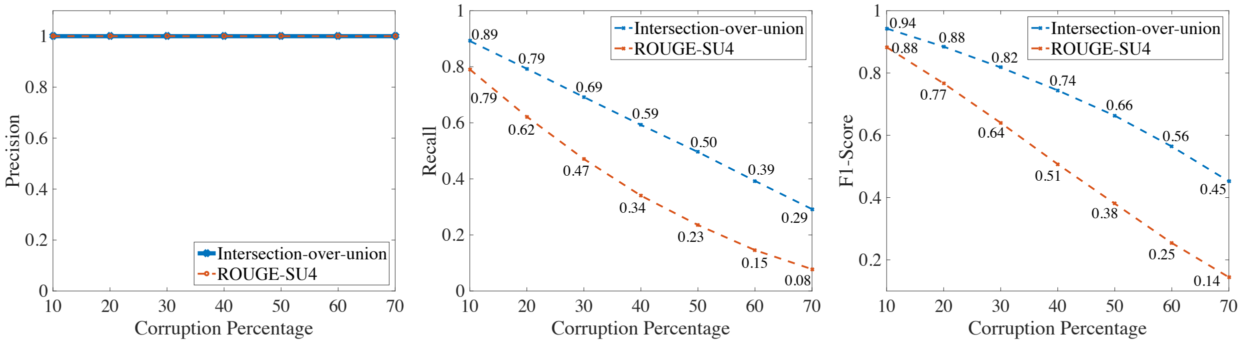


Figure 4.7: A nice behavior of our evaluation metric. When we randomly *remove* video shots from the user summaries, the recall between the original user summaries and the corrupted ones decreases almost linearly. The evaluation by ROUGE-SU4 [2] is included for reference.

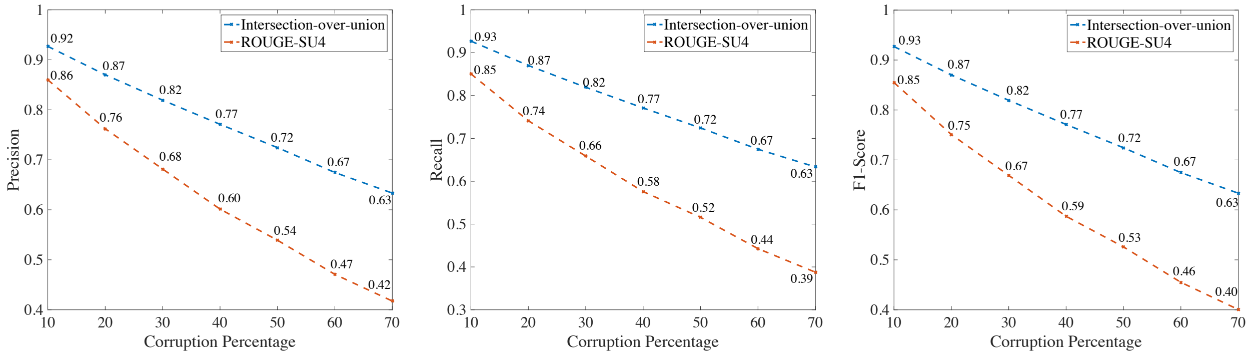


Figure 4.8: Studying the effect of randomly *replacing* some video shots in the user summary on the performance. The evaluation by ROUGE-SU4 [3] is included for reference.

We rectify those caveats by instead collecting dense concept annotations. Figure 4.2 exhibits a few video shots where the concepts we collected provide a better coverage than the captions about the semantics in the shots. Moreover, we conveniently define an evaluation metric based on the IOU similarity function between any two shots (cf. Section 4.1.2.1) thanks to the concept annotations.

Our evaluation metric has some nice behaviors. If we randomly remove some video shots from the user summaries and compare the corrupted summaries with the original ones, an accuracy-like metric should give rise to linearly decreasing values. This is indeed what happens to our recall as shown in Figure 4.7. In contrast, the ROUGE-SU4 recall, taking as input the shot captions, exhibits some nonlinearality.

4.4 Summary

In this Chapter, we studied the *subjectiveness* in video summarization. On our course to find a solution, we compiled a dataset that is densely annotated with a comprehensive set of concepts and designed a novel evaluation metric that benefits from the collected annotations. We also devised a new approach to generating personalized summaries by taking user queries into account.

We employed memory networks and determinantal point processes in our summarizer, so that our model leverages their attention schemes and diversity modeling capabilities, respectively. Extensive experiments verify the effectiveness of our approach and reveals some nice behaviors of our evaluation metric.

In the next Chapter, we enhance the DPP-based models from two different perspectives to achieving better summarization models. First, we develop a large-margin algorithm that enables more effective training of such models. Second, we design a new distribution that allows the resulting model to accept expected summary length from the users.

CHAPTER 5: GENERALIZED DPP AND A LARGE-MARGIN OBJECTIVE

The results of this Chapter have been published in the following paper:

Aidean Sharghi, Ali Borji, Chengtao Li, Tianbao Yang, and Boqing Gong, *"Improving Sequential Determinantal Point Processes for Supervised Video Summarization,"* in European Conference on Computer Vision, 2018, pp. 517-533.[4]

This Chapter 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 hours of videos in total) and compare our approach to several competitive baselines.

5.1 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 [1] and then describe the large-margin algorithm in detail.

5.1.1 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:

$$\hat{\boldsymbol{x}}_1 = \text{argmax}_{\boldsymbol{x} \in \mathcal{V}_1} P(X_1 = \hat{\boldsymbol{x}}), \quad \hat{\boldsymbol{x}}_2 = \text{argmax}_{\boldsymbol{x} \in \mathcal{V}_2} P(X_2 = \hat{\boldsymbol{x}} | X_1 = \hat{\boldsymbol{x}}_1), \quad ... \tag{5.1}$$

We note that, in the inference phase, a possible error at one time step (e.g., $\hat{\boldsymbol{x}}_1$) propagates to the future but MLE always feeds the oracle summary to SeqDPP in the training stage (i.e., exposure bias [45]). 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 [45]).

The issues above are common in seq2seq learning. It has been shown that improved results can be achieved if one tackles them explicitly [96, 97, 98, 45, 99]. 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 [46], which studies the large-margin principle in training recurrent neural networks to learn global sequence scores. propose a training procedure, inspired by the learning as search optimization (LaSO) framework of Daumé III and Marcu [100], that defines a loss function in terms of errors made during beam search. Furthermore, we provide an efficient algorithm to backpropagate through the beam-search procedure during seq2seq training.

However, unlike them, 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,

$$\mathcal{L}(\theta)=\sum_{t=1}^{T}\delta(x^*_{1:t-1}\cup\hat{x}_t,x^*_{1:t})M(x^*_t,\hat{x}_t,x^*_{t-1};\boldsymbol{L}), \tag{5.2}$$

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,

$$\begin{aligned}M(x^*_t,\hat{x}_t,x^*_{t-1};\boldsymbol{L}) :=&[1-\log P(X_t=x^*_t|x^*_{t-1})+\log P(X_t=\hat{x}_t|x^*_{t-1})]_+\\ =&[1-\log\det(\boldsymbol{L}_{x^*_t\cup x^*_{t-1}})+\log\det(\boldsymbol{L}_{\hat{x}_t\cup x^*_{t-1}})]_+\end{aligned} \tag{5.3}$$

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.

5.2 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.

5.2.1 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 $\mathcal{Y}$ [47]. To see this point more clearly, we can re-write a DPP as a mixture of elementary DPPs $P_E(Y)$ [101, Lemma 2.6],

$$P_L(Y;\boldsymbol{L}) = \frac{1}{\det(\boldsymbol{L}+\boldsymbol{I})}\sum_{J\subseteq\mathcal{Y}} P_E(Y;J)\prod_{n\in J}\lambda_n, \propto \sum_{k=0}^{\mathsf{N}}\sum_{J\subseteq\mathcal{Y},|J|=k} P_E(Y;J)\prod_{n\in J}\lambda_n \tag{5.4}$$

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 $\boldsymbol{L} = \sum_{n=1}^{\mathsf{N}}\lambda_n\boldsymbol{v}_n\boldsymbol{v}_n^T$, the marginal kernel of the elementary DPP $P_E(Y;J)$ is $\boldsymbol{K}^J = \sum_{n\in J}\boldsymbol{v}_n\boldsymbol{v}_n^T$ — 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, $P(|Y| = |J|) = 1$.

Eq. (5.4) 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 $\boldsymbol{\pi} = \{\pi_k\}_{k=0}^{\mathsf{N}}$ over the sizes of the subsets,

$$P_G(Y; \boldsymbol{L}) \propto \sum_{k=0}^{\mathsf{N}} \pi_k \sum_{J \subseteq \mathcal{Y}, |J|=k} P(Y; J) \prod_{n \in J} \lambda_n \tag{5.5}$$

As a result, the generalized DPP (*G*DPP) $P_G(Y; \boldsymbol{L})$ entails both DPP and k-DPP [47] as special cases (when $\boldsymbol{\pi}$ is uniform and when $\boldsymbol{\pi}$ is a Dirac delta distribution, respectively), offering a larger expressive spectrum. Another interesting result is that, for a truncated uniform distribution $\boldsymbol{\pi}$ over the sizes of the subsets, we arrive at a DPP which selects subsets with bounded cardinality, $P(Y \mid k_1 \le |Y| \le k_2; \boldsymbol{L})$. Such constraint arises from real applications like document summarization, image display, and sensor placement.

5.2.1.1 Normalization

To compute the normalization constant Z_G for GDPP, we have to sum over all the possible subsets of the ground set $\boldsymbol{y} \subseteq \mathcal{Y}$:

$$\begin{aligned} Z_G &\triangleq \sum_{\boldsymbol{y}} \sum_{k=0}^{\mathsf{N}} \pi(\kappa = k) \sum_{J \subseteq \mathcal{Y}, |J|=k} P(Y = \boldsymbol{y}; V_J) \prod_{n \in J} \lambda_n = \sum_{k=0}^{\mathsf{N}} \pi(\kappa = k) \sum_{|J|=k} \sum_{\boldsymbol{y}} P(Y = \boldsymbol{y}; V_J) \prod_{n \in J} \lambda_n \\ &= \sum_{k=0}^{\mathsf{N}} \pi(\kappa = k) \sum_{|J|=k} \prod_{n \in J} \lambda_n \triangleq \sum_{k=0}^{\mathsf{N}} \pi(k)\, e_{\mathsf{N}}(k), \end{aligned} \tag{5.6}$$

where $\pi(k) \triangleq \pi(\kappa = k)$, and $e_{\mathsf{N}}(k) \triangleq \sum_{|J|=k} \prod_{n \in J} \lambda_n$ is the elementary symmetric polynomial over the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{\mathsf{N}}$ of the DPP kernel $\boldsymbol{L}$.

Therefore, the computational complexity of the normalization constant Z_G for GDPP depends on the choice of the "size" distribution $\pi(k)$ and the complexity to compute $e_{\mathsf{N}}(k)$. Evaluating $\pi(k)$ usually takes less than $O(\mathsf{N}^2)$ time for $\pi(\kappa)$ being any popular discrete distribution. By Newton's identities or the recursive algorithm [101, Algorithm 7] we can compute all the elementary symmetric polynomials $e_{\mathsf{N}}(k), k = 1, 2, \cdots \mathsf{N}$, in $O(\mathsf{N}^2)$ time. Overall, the normalization complexity of GDPP hinges on the eigen-decomposition of $\boldsymbol{L}$ for obtaining λ_n in eq. (5.6), and is about the same as the complexity of normalizing an L-ensemble DPP (i.e., computing $\det(\boldsymbol{L}+\boldsymbol{I})$).

5.2.1.2 Evaluation

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

$$P_G(Y = \boldsymbol{y}; \boldsymbol{L}) = \frac{\pi_{|\boldsymbol{y}|}}{Z_G} \det(\boldsymbol{L}_{\boldsymbol{y}}) \tag{5.7}$$

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

5.2.1.3 GDPP as a Mixture of k-DPPs

The *G*DPP expressed above has a close connection to the k-DPPs [47]. This is not surprising due to the definition of *G*DPP (cf. Eq. (5.5)). Indeed, *G*DPP can be exactly interpreted as a mixture of $\mathsf{N}+1$ k-DPPs $P_k(Y = \boldsymbol{y}; \boldsymbol{L}), k = 0, 1, \cdots, \mathsf{N}$,

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

if ***all the k-DPPs***, i.e., the mixture components, ***share the same L-ensemble kernel $\boldsymbol{L}$*** as *G*DPP. 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 *G*DPP can then be written as

$$P_G(Y; \boldsymbol{L}) = \sum_{k=0}^{\mathsf{N}} p_k P_k(Y; \boldsymbol{L}). \tag{5.8}$$

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 *G*DPP of form (5.8) in the next section.

5.2.1.4 Exact Sampling

Following the interpretation of *G*DPP as a weighted combination of k-DPPs, we have the following decomposition of the probability:

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

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

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

5.2.2 Sampling via Eigendecomposition

Since GDPP could be viewed as a weighted mixture of k-DPP, we have the following decomposition of the probability:

$$P(S|S \sim \text{GDPP}) = P(S|S \sim k\text{-DPP})P(k|k \sim \text{GDPP}) \tag{5.9}$$

Here with a slight abuse of notation we let k $\sim$GDPP to denote the probability of sampling a k-DPP from GDPP. With aforementioned properties we have

$$P(k|k \sim \text{GDPP}) \propto \pi(k)e_k(L) \tag{5.10}$$

which can be computed in $\mathcal{O}(Nk + k^2)$ if we already have access to eigenvalues of L.

Hence, we can employ a 2-phase procedure of first deciding which k-DPP to sample from, and then sampling from corresponding k-DPP. The procedure is shown in Algorithm 2.

Algorithm 2 Sampling from GDPP via Eigendecomposition

```
Require: $L$ the GDPP kernel, $\pi$ the (unnormalized) size probability, $k^{\max}$ and $k^{\min}$ as stated in
   the assumption, $V$ the ground set
Ensure: $S$ sampled from GDPP
   Sample $k$ from discrete distribution $\{p_i = \pi(i)e_i(L)\}_{i=0}^{N}$
   Sample $S$ from $k$-DPP
```

5.2.3 Sampling via Markov Chain

We consider sampling from GDPP via Markov chains. While running the chain, we maintain a currently active set as the current state for the chain, and in each iteration we try to update the current active set only slightly with certain probabilities such that the update is efficient to be done and the stable distribution – the distribution of active set when running chain long enough – is the same as GDPP. Since in each iteration we only update the active set slightly, the probability of transferring from one state to any other states in the chain is non-zero. To fulfill this condition, it suffices to assume that $\pi_i > 0$ for all i such that, there exists $j < i$ and $k > i$ such that $\pi_j > 0$ and $\pi_k > 0$.

The sampling efficiency of Markov chain heavily depends on its mixing time. Thus, besides constructing the chain, we also show that the constructed chains are fast mixing.

We construct an add-delete Markov chain to sample from GDPP. Concretely, we update the active set by either trying to add an element to or delete an element from it with certain transition probabilities. The full algorithm is shown in Algorithm 3.

Algorithm 3 Add-Delete Markov Chain for GDPP

```
Require: $L$ the GDPP kernel, $\pi$ the (unnormalized) size probability, $k^{\max}$ and $k^{\min}$ as stated in
  the assumption, $V$ the ground set
Ensure: $S$ sampled from GDPP
  Initialize $S$ s.t. $P(S) > 0$
  while not mixed do
      Let $b = 1$ with probability $\frac{1}{2}$
      if $b = 1$ then
          Pick $s \in V$ uniformly randomly
          if $s \notin S$ and $|S| < k^{\max}$ then
              $S \leftarrow S \cup \{s\}$ with probability $p^{+}(S, s) = \frac{F(S\cup\{s\})}{F(S)+F(S\cup\{s\})}$
          else if $s \in S$ and $|S| > k^{\min}$ then
              $S \leftarrow S \backslash \{s\}$ with probability $p^{-}(S, s) = \frac{F(S\backslash\{s\})}{F(S)+F(S\backslash\{s\})}$
      else
          Do nothing
```

5.2.3.1 Mixing Time

To show fast mixing, we consider using *path coupling*, which essentially says that if we have a contraction of two (coupling) chains then we have fast mixing. Assume we have a chain (S_t) on state space 2^V with transition matrix P, a *coupling* is a new chain (S_t, Y_t) on $V \times V$ such that both (S_t) and (Y_t), if considered marginally, are Markov chains with the same transition matrices P. The key point of coupling is to construct such a new chain to encourage S_t and Y_t to *coalesce* quickly. If, in the new chain, $\Pr(S_t \neq Y_t) \leq \varepsilon$ for some fixed t regardless of the starting state (S_0, Y_0), then $\tau(\varepsilon) \leq t$ [102]. To make the coupling construction easier, *Path coupling* [103] is then introduced so as to reduce the coupling to adjacent states in an appropriately constructed state graph. The coupling of arbitrary states follows by aggregation over a path between the two. Path coupling is formalized in the following lemma.

Lemma 1. *[103, 104] Let δ be an integer-valued metric on $V \times V$ where $\delta(\cdot,\cdot) \leq D$. Let E be a subset of $V \times V$ such that for all $(S_t, Y_t) \in V \times V$ there exists a path $S_t = Z^0, \ldots, Z^r = Y_t$ between S_t and Y_t where $(Z^i, Z^{i+1}) \in E$ for $i \in [r-1]$ and $\sum_i \delta(Z^i, Z^{i+1}) = \delta(S_t, Y_t)$. Suppose a coupling $(S,T) \to (S',T')$ of the Markov chain is defined on all pairs in E such that there exists an $\alpha < 1$ such that $\mathbb{E}[\delta(S',T')] \leq \alpha\delta(S,T)$ for all $(S,T) \in E$, then we have $\tau(\varepsilon) \leq \frac{\log(D\varepsilon^{-1})}{(1-\alpha)}$.*

With path coupling we are able to bound the mixing time of Algo. 3 as follows.

Theorem 1. *Let $\alpha = \max_{(S,T)\in E}\{\alpha_1, \alpha_2\}$ where α_1 and α_2 are defined as $\llbracket$*

$$\begin{aligned}
\alpha_1 =& 1 - \llbracket |T| > k^{\min} \rrbracket \sum_{i\in T} |p^-(T,i) - p^-(S,i)|_+ - \llbracket |S| < k^{\max} \rrbracket \sum_{i \in V\setminus S} |p^+(S,i) - p^+(T,i)|_+; \\
\alpha_2 =& \llbracket |S| > k^{\min} \rrbracket (\min\{p^-(S,s), p^-(T,t)\} - \sum_{i\in R} |p^-(S,i) - p^-(T,i)|) + \\
& \llbracket |S| < k^{\max} \rrbracket (\min\{p^+(S,t), p^+(T,s)\} - \sum_{i\in V\setminus(S\cup T)} |p^+(S,i) - p^+(T,i)|).
\end{aligned}$$

In the expression, summations over absolute difference quantifies the sensitivity of transition prob-

abilities to adding/deleting elements in neighboring (S,T) *in* E*. Assuming* $\alpha < 1$*, we have*

$$\tau(\varepsilon) \leq \frac{2N\log(k^{\max}\varepsilon^{-1})}{1-\alpha}$$

Proof. We define $\delta(X,Y) = \frac{1}{2}(|X \oplus Y| + ||X| - |Y||)$. It is clear that $\delta(X,Y) \geq 1$ for $X \neq Y$. Let $E = \{(X,Y) : \delta(X,Y) = 1\}$ be the set of adjacent states (neighbors), and it follows that $\delta(\cdot,\cdot)$ is a metric satisfying conditions in Lemma 1. Also we have $\delta(X,Y) \leq k^{\max}$.

We consider constructing a path coupling between any two states S and T with $\delta(S,T) = 1$, S' and T' be the two states after transition. We sample $c_S, c_T \in \{0,1\}$, if c_S is 0 then $S' = S$ and the same with c_T. $i_S, i_T \in V$ are drawn uniformly randomly. We consider two possible settings for S and T:

1. If S or T is a subset of the other, we assume without of generality that $S = T \cup \{t\}$. In this setting we always let $i_S = i_T = i$. Then

   (a) If $i = t$, we let $c_S = 1 - c_T$;

      i. If $c_S = 1$ then $\delta(S',T') = 0$ with probability $p^-(S,t)$;

      ii. If $c_S = 0$ then $\delta(S',T') = 0$ with probability $p^+(T,t)$;

   (b) If $i \in T$, we set $c_S = c_T$;

      i. If $c_S = 1$ and $|T| > k^{\min}$ then $\delta(S',T') = 2$ with probability $(p^-(T,i) - p^-(S,i))_+$;

   (c) If $i \in V\backslash S$, we set $c_S = c_T$;

      i. If $c_S = 1$ and $|S| < k^{\max}$ then $\delta(S',T') = 2$ with probability $(p^+(S,i) - p^+(T,i))_+$.

2. If S and T are of the same sizes, let $S = R \cup \{s\}$ and $T = R \cup \{t\}$. In this setting we always let $c_S = c_T = c$. We consider the case of $c = 1$:

   (a) If $i_S = s$, let $i_T = t$. If $|S| > k^{\min}$, then $\delta(S', T') = 0$ with probability $\min\{p^-(S, s), p^-(T, t)\}$;

   (b) If $i_S = t$, let $i_T = s$. If $|S| < k^{\max}$, then $\delta(S', T') = 0$ with probability $\min\{p^+(S, t), p^+(T, s)\}$;

   (c) If $i_S \in R$, let $i_T = i_S$. If $|S| > k^{\min}$, then $\delta(S', T') = 2$ with probability $|p^-(S, i_S) - p^-(T, i_T)|$;

   (d) If $i_S \in V \backslash (S \cup T)$, let $i_T = i_S$. If $|S| < k^{\max}$, then $\delta(S', T') = 2$ with probability $|p^+(S, i_S) - p^+(T, i_T)|$.

In all cases where we didn't specify $\delta(S', T')$, it will be $\delta(S', T') = 1$. In the first case of $S = T \cup \{t\}$ we have

$$\begin{aligned}\frac{\mathbb{E}[\delta(S', T')]}{\mathbb{E}[\delta(S, T)]} \leq & \frac{1}{2N}((1 - p^-(S, t)) + (1 - p^+(T, t)) + \\ & (2|T| + [\![|T| > k^{\min}]\!] \sum_{i \in T} (p^-(T, i) - p^-(S, i))_+) + \\ & (2(N - |S|) + [\![|S| < k^{\max}]\!] \sum_{i \in V \backslash S} (p^+(S, i) - p^+(T, i))_+)) \\ = 1 - \frac{1}{2N}(1 - & [\![|T| > k^{\min}]\!] \sum_{i \in T} (p^-(T, i) - p^-(S, i))_+ \\ & - [\![|S| < k^{\max}]\!] \sum_{i \in V \backslash S} (p^+(S, i) - p^+(T, i))_+) = 1 - \frac{\alpha_1}{2N},\end{aligned}$$

while in the second case of $|S| = R \cup \{s\}$ and $T = R \cup \{t\}$ we have

$$\begin{aligned}\frac{\mathbb{E}[\delta(S',T')]}{\mathbb{E}[\delta(S,T)]} \leq& \frac{1}{2N}((1 - [\![|S| > k^{\min}]\!] \min\{p^-(S,s), p^-(T,t)\})+ \\ & (1 - [\![|S| < k^{\max}]\!] \min\{p^+(S,t), p^+(T,s)\})+ \\ & (2|R| + [\![|S| > k^{\min}]\!] \sum_{i \in R} |p^-(S,i) - p^-(T,i)|)+ \\ & (2(N - |S| - 1) + [\![|S| < k^{\max}]\!] \sum_{i \in V \setminus (S \cup T)} |p^+(S,i) - p^+(T,i)|)) \\ =1 - \frac{1}{2N}&([\![|S| > k^{\min}]\!] \min\{p^-(S,s), p^-(T,t)\} - \sum_{i \in R} |p^-(S,i) - p^-(T,i)|+ \\ & [\![|S| < k^{\max}]\!](\min\{p^+(S,t), p^+(T,s)\}- \\ & \sum_{i \in V \setminus (S \cup T)} |p^+(S,i) - p^+(T,i)|)) = 1 - \frac{\alpha_2}{2N}.\end{aligned}$$

Let $\alpha = \max_{(S,T) \in E}\{\alpha_1, \alpha_2\}$. If $\alpha < 1$, with Lemma 1 we have

$$\tau(\varepsilon) \leq \frac{2N \log(k^{\max}/\varepsilon)}{1 - \alpha}.$$

□

5.2.3.2 *Remarks*

This bound involves some constants that is hard to compute in practice. It is more general due to the generality of GDPP – when setting π to be uniform in $[k]$ and set $k^{\min} = 0$, $k^{\max} = k$, we recover the bound in [105].

5.2.4 A Sequential Model of GDPPs (SeqGDPP)

In this section, we construct a sequential model of the generalized DPPs (Seq*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 $\bigcup_{t=1}^{\mathsf{T}} \mathcal{V}_t = \mathcal{V}$. The main idea of Seq*G*DPP is to adaptively distribute the expected length M_0 of the video summary to different video segments over each of which a *G*DPP is defined. In particular, we replace the conditional DPPs in SeqDPP (cf. eq. (2.3)) by *G*DPPs,

$$P(X_t = \boldsymbol{x}_t | X_{t-1} = \boldsymbol{x}_{t-1}) \tag{5.11}$$

$$\triangleq P_G(X_t = \boldsymbol{x}_t; \boldsymbol{\Omega}^t) = p^t_{|\boldsymbol{x}_t|} P_{|\boldsymbol{x}_t|}(X_t = \boldsymbol{x}_t; \boldsymbol{\Omega}^t), \tag{5.12}$$

where the last equality follows Eq. (5.8), and recall that the L-ensemble kernel $\boldsymbol{\Omega}^t$ encodes the dependencies on the video frames/shots selected from the immediate past segment $\boldsymbol{x}_{t-1} \subseteq \mathcal{V}_{t-1}$ (cf. Section 2.6.1, Eq. (2.4)). The discrete distribution $\boldsymbol{p}^t = \{p^t_k\}$ is over all the possible sizes $\{k\}$ of the subsets at time step t.

We update $\boldsymbol{p}^t$ adaptively according to

$$p^t_k \propto \exp(-\alpha(k - \mu^t)^2), \tag{5.13}$$

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 *G*DPP $P_G(X_t; \boldsymbol{\Omega}^t)$ 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,

$$\mu^t \triangleq \frac{\mathsf{M}_0 - \sum_{t'=1}^{t-1} |\boldsymbol{x}_{t'}|}{\mathsf{T} - t + 1} + \boldsymbol{w}^T \phi(\mathcal{V}_t) \tag{5.14}$$

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 $\boldsymbol{w}^T\phi(\mathcal{V}_t)$ 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 $\boldsymbol{w}$ from the training data — user annotated video summaries and their underlying videos. We expect that a visually homogeneous video segment gives rise to negative $\boldsymbol{w}^T\phi(\mathcal{V}_t)$ such that less than the average number of items will be selected from it, and vice versa.

5.2.5 *Learning and Inference*

For the purpose of out-of-sample extension, we shall parameterize Seq*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 $X_{t-1} = \boldsymbol{x}_{t-1}$ from the immediate previous time step. We use a simple convex combination of D base *G*DPPs whose kernels are predefined over the video for the parameterization. Concretely, at each time step t,

$$\begin{aligned} P(X_t|\boldsymbol{x}_{t-1}, \mathcal{V}_t) =& P_G(X_t; \boldsymbol{\Omega}^t, \mathcal{V}_t) \triangleq \sum_{i=1}^{\mathsf{D}} \beta_i P_G(X_t; \boldsymbol{\Omega}^{t(i)}, \mathcal{V}_t) \\ =& \sum_{k=0}^{|\mathcal{V}_t|} p_k^t \sum_{i=1}^{\mathsf{D}} \beta_i P_k(X_t; \boldsymbol{\Omega}^{t(i)}, \mathcal{V}_t) \end{aligned} \tag{5.15}$$

where the L-ensemble kernels $\boldsymbol{\Omega}^{t(i)}, i = 1, \cdots, \mathsf{D}$ of the base *G*DPPs are derived from the corresponding kernels $\boldsymbol{L}^{t(i)}$ of the conditional DPPs (eq. (2.4)). We compute different Gaussian RBF kernels for $\boldsymbol{L}^{t(i)}$ from the segment $\mathcal{V}_t$ and previously selected subset $\boldsymbol{x}_{t-1}$ 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 $\mathcal{V} = \cup_{t=1}^{\mathsf{T}} \mathcal{V}_t$ and its user summary $\{\boldsymbol{x}_t \subseteq \mathcal{V}_t\}_{t=1}^{\mathsf{T}}$ for the convenience of presentation. We learn Seq*G*DPP by maximizing the log-likelihood,

$$\mathcal{L} = \log \text{Seq}G\text{DPP} = \sum_{t=1}^{\mathsf{T}} \log P(X_t = \boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \mathcal{V}_t)$$

$$= \sum_{t=1}^{\mathsf{T}} \log p_{|\boldsymbol{x}_t|}^{t} + \sum_{t=1}^{\mathsf{T}} \log \left(\sum_{i=1}^{\mathsf{D}} \beta_i P_{|\boldsymbol{x}_t|} \Big(X_t = \boldsymbol{x}_t ; \Omega_i^{t(i)} \Big) \right).$$

5.3 Experiments

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.

5.3.1 Dataset

While various video summarization datasets exist [24, 19, 94], 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 [33] dataset contains 4 videos each between 3∼5 hours long, covering activities such as driving, shopping, studying, etc. in uncontrolled environments. We build our video summarization dataset by extending it with another 8 egocentric videos (on average over 6 hours long each) from the social interactions dataset [48]. 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 hours.

5.3.2 User Summary Collection

We recruit three students to summarize the videos. The only instruction we give them is to operate on the 5-second 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 5.1 exhibit that the users have their own distinct preferences about the summary lengths.

Table 5.1: 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

5.3.3 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 [1, 38, 39], and learn using them as the supervision.

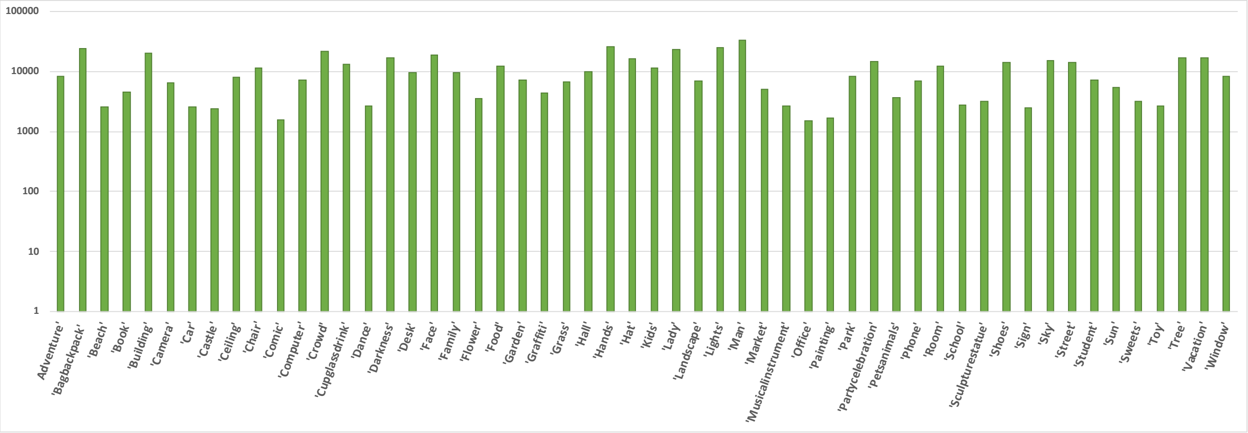


Figure 5.1: Count of concept appearances in the collected annotations for the 12 videos.

5.3.4 Features

We follow Zhang et al. [36] in extracting the features, i.e., using a pre-trained GoogLeNet [106] to obtain the frame's pool5 activations and then aggregating them to a 1024-d feature representation for each shot of the video.

5.3.5 Evaluation

In the previous Chapter, we 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.) and use them to measure the quality of system summaries from the semantics perspective. 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 [39] and remove the concepts that do not appear frequently enough such as BOAT and OCEAN. Furthermore, we apply SentiBank detectors [107] (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. Figure 5.1).

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 $\sim$11 concepts. This is significantly larger than the average of 4 tags/shot we had obtained previously, resulting in more reliable assessment upon evaluation. Amazon Mechnical Turk. Figure 5.1 shows the total number of each visual concept appeared in our dataset.

While the metric introduced in [39] 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 5.2.

5.3.6 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 hyper-parameters), and the remaining 9 for training. We report the average results of the 12 rounds of experiments.

5.3.7 Large-Margin Training and Inference

Similar to the practices in seq2seq learning [46, 45], 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. (5.1) to generate the system summary.

5.3.8 SeqGDPP Details

Given the features that are extracted using GoogLeNet, we compute the Gaussian RBF kernels $\{\boldsymbol{L}^{t(i)}\}_{i=1}^{\mathsf{D}}$ 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 $\{\boldsymbol{\Omega}^{t(i)}\}$ for *G*DPPs are then computed through eq. (2.4) 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. (5.14), we use such feature vector to help determine the mean of the distribution $\boldsymbol{p}$ 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 $\boldsymbol{w}$ in the distribution over the subset size, and $\{\beta_i\}$ for the convex combination of some base *G*DPPs. We consider $\boldsymbol{w}$ 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.

5.3.9 Computational Cost Comparison

It takes about 28 seconds for SeqDPP to complete one epoch of the MLE training and about 4 seconds 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.

5.3.10 Quantitative Results and Analyses

5.3.10.1 Generic Video Summarization

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 5.2: Comparison results for supervised video summarization (%). The AUCs are computed by the F1-score curves drawn in Figure 5.2 until the 60 seconds mark. The blue and red colors group the base model and its large-margin version.

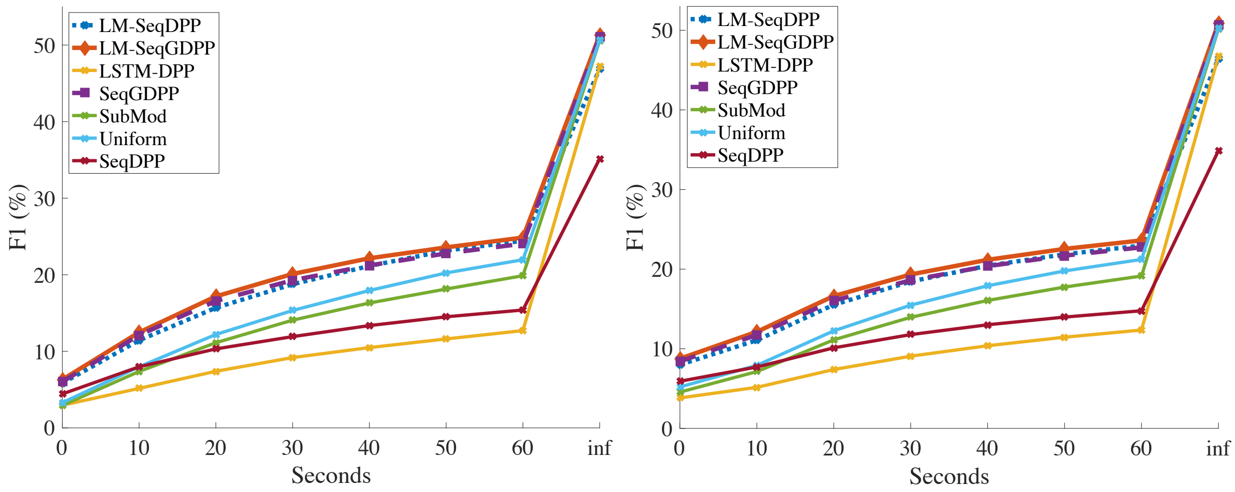
	AUC_{Π}	$\text{AUC}_{\text{Gaussian}}$
Uniform	12.33	12.36
SubMod [28]	11.20	11.12
SuperFrames [24]	11.46	11.28
LSTM-DPP [36]	7.38	7.36
SeqDPP [1]	9.71	9.56
LM-SeqDPP	15.05	14.69
SeqGDPP	15.29	14.86
LM-SeqGDPP	**15.87**	**15.43**

– *SubMod*. Gygli et al. [28] 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 [24], 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-second 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 [36], 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. [1]. Unlike other baselines, this model determines the summary length automatically.



(a) Π temporal filter

(b) Gaussian temporal filter

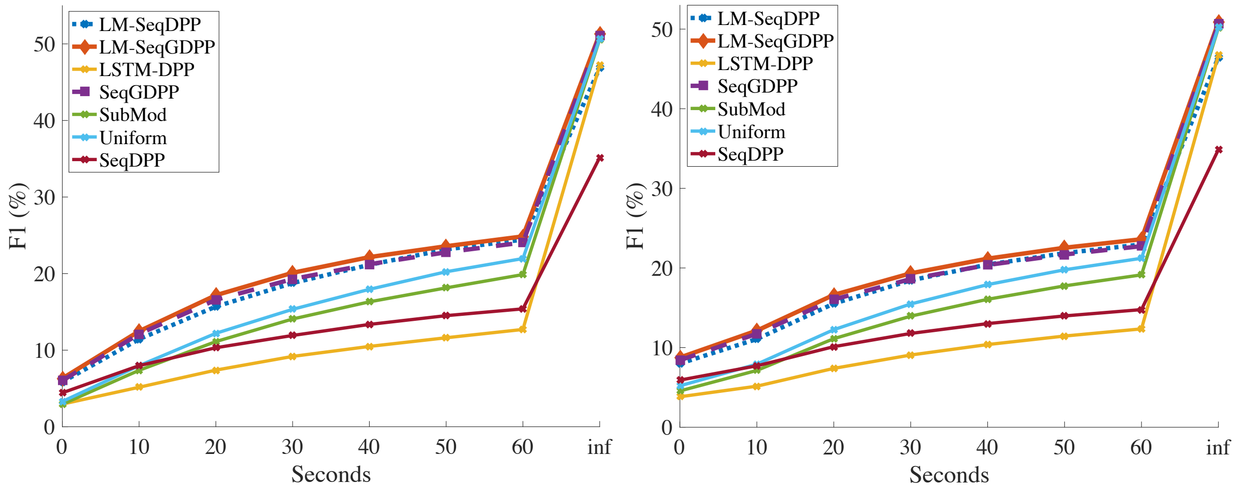
Figure 5.2: Comparison results for supervised video summarization. The X axis represents the temporal filters' parameters. In the case of the Π filter, it indicates how far apart a match can be temporally (in terms of seconds), whereas in the Gaussian filter, it is the kernel bandwidth.

The comparison results are shown in Table 5.2 and Figure 5.2. 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 Figure(5.2), 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.



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

Figure 5.3: Comparison results for query-focused video summarization task. x axis represent the temporal filter parameter. In case of Π filter, it indicates how far a match can be temporally (in terms of seconds), whereas in the Gaussian filter, it is the kernel bandwidth.

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 [38] — 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 Figure 5.2 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 Figure(5.2)).

5.3.10.2 Query-Focused Video Summarization

In Figure(5.3) and Table(5.3), we compare the baselines with Seq*G*DPP and large-margin modification of SeqDPP and MemNet. Comparing base SeqDPP with LM-SeqDPP, the performance gain is quite significant. On the other hand, MemNet preforms competitive to LM-MemNet. LM-Seq*G*DPP outperforms all the models, and this is in line with our generic video summarization results reported in Table(5.2).

Table 5.3: Comparison results for query-focused video summarization (%). AUCs are computed on the curves drawn in Figure(5.3) until the 60 seconds mark. Matching colors indicate the base-model and its equivalent large-margin peer.

	AUC_{Π}	$\text{AUC}_{\text{Gaussian}}$
SH-DPP [38]	7.03	6.92
LSTM-DPP [36]	5.99	5.97
SeqDPP [1]	6.14	6.18
LM-SeqDPP	9.70	9.66
MemNet [39]	9.93	9.77
LM-MemNet	9.83	9.67
SeqGDPP	9.72	9.66
LM-SeqGDPP	10.43	**10.30**

5.4 Summary

In this Chapter, we made twofold contribution towards improving the sequential determinantal point process (SeqDPP) models for supervised video summarization. We proposed 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 introduced a new probabilistic module, GDPP, which enables the resulting sequential model to accept priors about the expected summary length. Finally, we compiled a large video summarization dataset consisting of 12 egocentric videos totalling over 60 hours. We collected 3 user-annotated summaries per video as well as dense concept annotations required for the evaluation. Experiments on this dataset verified the effectiveness of our large-margin training algorithm as well as the sequential GDPP model.

In the next Chapter, we develop a novel framework to generate text synopsis for a given video. Similar to conventional methods, the input to our model is a video, however the main output in our system is a short textual summary. This summary consists of several sentences that are chosen based on their individual naturalness as well as their significance in conveying important information about the video when considered together.

CHAPTER 6: TEXT SYNOPSIS GENERATION FOR VIDEOS

The content of this Chapter have been submitted to Computer Vision and Pattern Recognition (CVPR 2020) for review.

Conventionally, video summarization techniques create a summary video that only includes important frames/shots of the original video in temporal order and hence are significantly shorter. While this can be helpful, it is rather ineffective for browsing large databases as one still has to watch the summaries to infer information about the original videos. Hence, in this Chapter, we propose to generate a **textual synopsis**, consisting of a few sentences describing the most important events in a long egocentric videos. Users can read the short text to gain insight about the video, and more importantly, efficiently search through the content of a large video database using text queries. Since egocentric videos are long and contain many activities and events, using video-to-text algorithms results in thousands of descriptions, many of which are incorrect. Therefore, we propose a multi-task learning scheme to simultaneously generate descriptions for video segments and summarize the resulting descriptions in an end-to-end fashion. We Input a set of video shots and the network generates a text description for each shot. Next, **visual-language content matching unit** that is trained with a **weakly** supervised objective, identifies the correct descriptions. Finally, the last component of our network, called **purport** network, evaluates the descriptions all together to select the ones containing crucial information. Out of thousands of descriptions generated for the video, a few informative sentences are returned to the user. We validate our framework on the challenging UT Egocentric video dataset, where each video is between 3 to 5 hours long, associated with over 3000 textual descriptions on average. The generated textual summaries, including only 5 percent (or less) of the generated descriptions, are compared to groundtruth summaries in text domain using well-established metrics in natural language processing.

6.1 Methodology

Given a long egocentric video, our main goal is to produce a textual summary or report of it that consists of sentences. To this end, we propose an end-to-end multi-task learning scheme to simultaneously densely caption the video and summarize the large pool of generated captions. These tasks, when learned together, allows us to identify which video shots do not yield correct captions, hence significantly boosting the chance of including correct descriptions in the eventual textual summary. We explain components of our model in details in the remainder of this section.

6.1.1 Naïve Text Synopsis Generation

One straightforward to generating text synopsis for a video is to first select important/interesting shots in the video using an off-the-shelf video summarization algorithm. Having selected a set of shots, one can feed each shot to a pre-trained video-to-text network, also known as video caption generator, to generate a description for it. By putting all the generated descriptions into a document, one obtains the textual synopsis associated with the video.

There are two major flaws in models following this architecture. Firstly, since the video summarization algorithm merely considers the low-level visual features of a video shot to measure its interestingness, it may select shots that are visually different, but represent the similar activity or event. Similar descriptions are generated by the captioning network for such similar shots, hence, resulting in a textual summary with redundancy of information. Secondly, even if the video summarizer chooses perfect shots, generating correct descriptions for long egocentric videos is extremely challenging. Therefore, if not dealt with, some descriptions included in the textual summary of the video could be wrong. These flaws result in suboptimal text synopses.

Another approach to generating text synopsis for a video is to first generate dense captions using existing video-to-text algorithms to obtain a long text describing all events in the video, important or not. Next, a text summarization model can be employed to summarize this long document by selecting a few most representative sentences. The major flaw in such models originates from the disconnect between the captioning network and the text summarization unit. The captioning module will generate many descriptions (thousands per video in our experiments), some of which may be meaningless or incorrect. However, the text summarization algorithm works under the premise that the given document is minimally noisy. As we will demonstrate in later, this assumption does not hold, especially when dealing with long egocentric videos.

To resolve the issues mentioned above, we propose a multi-task training scheme that joins the two tasks of caption generation and text summarization in an end-to-end fashion. This eliminates the disconnect between the two tasks and allows the model to both effectively densely caption the video and summarize the large pool of captions, taking into consideration that some may be wrong. To achieve this goal, first visual features of the video shots are fed to an LSTM-based video caption generation network to generate a textual description for each shot, in form of a natural language sentence. Next, each generated sentence is passed through a module that assigns to it a correctness score that measures how well it describes the visual content of the video shot. In parallel, another module processes all the generated sentences in temporal order, assigning a significance score to each based on their importance in the context of the video. Subsequently, summary-level impact of each sentence, represented by a scalar, is formulated as the product of its correctness and significance scores. This way the worthiness of a sentence based on its individual quality as well as the importance of the event that it is describing in the context of the video is evaluated. The summary-level impact values in the temporal order form a time series where the peaks correspond to locally (in time) important events in the video. Finally, the sentences with peak impact values are collected in a document in temporal order and this text synopsis to returned to the user.

6.1.2 Unified Text Synopsis Generation Framework

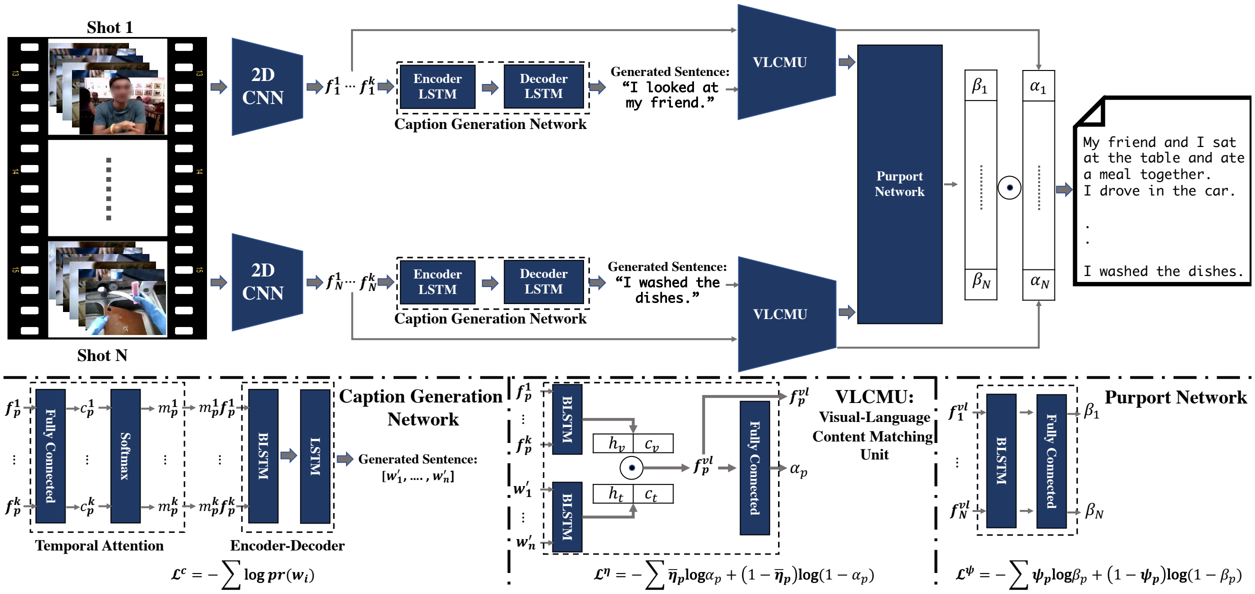


Figure 6.1: A complete overview of our proposed approach. Given a video, we partition it uniformly into shots, subsample k frames from each shot, and pass them to a convolutional neural network for feature extraction ($\{\boldsymbol{f}_p^1, \cdots, \boldsymbol{f}_p^k\}$, where p indicates the shot index). The set of features for each shot is then fed to a caption generation network that produces a sentence describing it. Our Visual-language Content Matching Unit (VLCMU) learns to assign a scalar value $\alpha_p \in [0, 1]$ to each description based on how correct it is in describing the video shot. In addition to α_p, VLCMU produces a *visual-language* feature representation, $\boldsymbol{f}_p^{vl}$, for a shot given its visual features and generated description. The feature representations of all shots in the video are then passed through our purport network. Unlike VLCMU that works on shot level, purport network performs at video level, taking all visual-language shot representations to assign a a scalar value $\beta_p \in [0, 1]$ to each, measuring importance of p^{th} shot's **description** in the context of the video. Finally, impact score of a description, γ_p, is computed as the product of its correctness and significance scores, $\alpha_p \times \beta_p$. The descriptions pertaining to peak significance scores are put in the text synopsis and are returned to the user.

We illustrate our complete framework for producing textual synopsis from videos in Figure 6.1. Given a video, first we partition it uniformly into non-overlapping shots due to three main reasons. First, the UT Egocentric dataset employed to validate our model on is annotated the same way. Yeung et al. [2] uniformly partitioned each video in UT Egocentric dataset into non-overlapping 5-sec long shots and collected a textual description for each. This allows us to effectively train and evaluate our model. Second, since almost all events in the videos are longer than shot length, splitting the events into short shots results in multiple visually similar clips that have same or similar descriptions. This allows us to effectively train our caption generation network (we need more than one sample per description to train a network that can generalize). Third, this is indeed the most efficient way to densely caption a long egocentric video. Any alternative approach requires either a proposal module (to propose video segments) or a shot boundary detector, both of which yield unnecessary complications in the model. It is also worth mentioning that even if the captioning network produces identical descriptions for consecutive shots, this does not reduce the performance. This is due to the fact that we remove duplicate sentences that appear consecutively during inference, all events (whether long or short) is described in a single short sentence. Hence, uniformly splitting the videos does not pose a problem.

Next, k frames are sampled from each shot's pool of frames. Each frame is then passed through a pre-trained CNN for feature extraction. Thus, each shot is represented by the set of its frame-level feature representations, i.e. $\{\boldsymbol{f}_p^1, \cdots, \boldsymbol{f}_p^k\}$, where $p \in \{1, \cdots, N\}$ indicates the shot index number. The rest of our pipeline consists of: 1) a caption generation network, 2) a visual-language matching unit, and 3) a purport network. In the following, we discuss the details of these submodules.

6.1.2.1 Video Caption Generation Network

Given a shot's feature set, we wish to generate a sentence that describes it. Since each shot is a short video clip, we develop our own caption generation network, instead of using an off-the-shelf model. The reason behind is that the complexity of these models mainly originate from a proposal unit that is unnecessary due to short length of the shots. To confirm this claim, we adopted the captioning network of Wang et al.'s [68] and performed two experiments. In our first experiment, we initialized their pre-trained (on ActivityNet Captions [67]) model, and fine-tuned it on UT

Egocentric [33]. This yielded significantly lower performance. The underlying reason behind such inferior performance is the significant vocabulary difference between the two datasets. In other words, existing captioning datasets are not suitable for captioning long egocentric videos and will not provide improvements. In a second experiment, we trained their model from scratch on UT Egocentric, and were only able to achieve comparable results. Therefore, we proceeded with our own design.

Detailed structure of our caption generation network is presented in bottom left corner of Figure 6.1. It consists of a temporal attention module, a bidirectional LSTM that serves as the encoder, and a decoder LSTM that generates the sentences.

More formally, given a feature set, $\{\boldsymbol{f}_p^1, \cdots, \boldsymbol{f}_p^k\}$, of a shot, each $\boldsymbol{f}_p^i$ is fed to fully connected layer with output dimension of 1, i.e. a scalar output c_p^i. A softmax activation is applied on top of $[c_p^1, \cdots, c_p^k]$ to transform them into probability values $[m_p^1, \cdots, m_p^k]$, that serve as weights for the original features. The purpose of the designated temporal attention module is to find most representative features of a shot and only use those for caption generation. This is a crucial step, specially when dealing with egocentric videos in which the camera is constantly moving. The temporal attention module discards uninformative feature representations to increase the quality of generated sentences.

We feed the weighted features to the encoder, that is a bidirectional LSTM, and obtain the cell and hidden states. These states are then used to initialize the decoder states which produces a sentence for each shot. Conventional sum of negative log likelihood of the words is used as the objective function to train our caption generation network:

$$\mathcal{L}_p^c = -\sum_{i=1}^{m} \log(pr(w_i)), \tag{6.1}$$

where m is the number of words in the groundtruth sentence, w_i is the i^{th} word in it and $pr(.)$ yields the likelihood of the input word. Note that $\mathcal{L}^c_p$ in Equation 6.1 is for p^{th} shot and the overall caption generation loss ($\mathcal{L}^c$) is computed by summing loss of all captions in training set.

6.1.2.2 Visual-Language Content Matching Unit

A well-trained caption generation network generates sentences that resemble sentences written by human subjects. For a long egocentric video, we generate a few **thousand** descriptions, however, not all of them are descriptive of their corresponding shots (i.e. the sentence is meaningful on its own but it is a wrong description for the given video clip). In fact, our study shows that only a small fraction of the generated sentences are informative. Thus, it is crucial that our model can distinguish between informative/uninformative sentences.

To recognize that a generated sentence is a wrong description for a given shot, only attending to the sequence of words in the sentence is no longer sufficient. These sentences resemble those written by human subjects, but do not describe the scene. To identify such cases, we must attend to both the visual features as well as the sequence of words in the generated sentence.

Formally, our Visual-Language Content Matching Unit (VLCMU) takes as input visual features $\{\boldsymbol{f}^1_p, \cdots ,\boldsymbol{f}^k_p\}$ of a shot, as well as its corresponding generated sentence $\{\boldsymbol{w}^{'}_1, \cdots , \boldsymbol{w}^{'}_n\}$ (where n is the number of words in the sentence and $w^{'}_i$ is the i^{th} word in p^{th} shot's sentence), and assigns to it a scalar score $\alpha_p \in [0, 1]$. Note that we omit subscript p in notation used for words in the sentence to enhance legibility. Ideally, this unit assigns scores close to 0 to uninformative and 1 to informative sentences respectively.

Our complete VLCMU network is shown in Figure 6.1 at the bottom. Two bi-directional LSTM units are employed, one processing the visual features of a shot and the other reads its corresponding generated sentence word-by-word. Hidden and cell states of each BLSTM unit are obtained and concatenated. Resulting representations are then elementwise multiplied to produce a *visual-language* feature representation, $\boldsymbol{f}_p^{vl}$, for that shot. Indeed visual and textual features belong to different feature spaces, however, LSTM 32 has been proven sufficient in learning the common embedding as well as processing the sequences at once [108]. Finally, we feed this new feature representation to a fully connected layer with sigmoid nonlinearity to obtain the correctness score α_p.

We use a binary-crossentropy loss $\mathcal{L}^\eta$ for this network that allows us to learn its parameters. For a generated sentences, we set its **pseudo groundtruth** correctness score $\bar{\eta}_p$ to 1 if more than half of its constituent words appear in the groundtruth sentence and otherwise to 0. This **weak** supervision effectively eliminates the need to annotate the generated sentences during every epoch of training. $\mathcal{L}^\eta$ is computed as follows:

$$\mathcal{L}^\eta = -\sum_{p=1}^{N} \bar{\eta}_p \log \alpha_p + (1 - \bar{\eta}_p) \log(1 - \alpha_p), \tag{6.2}$$

As shown in our ablation study (please refer to Table 6.4), the existence of this unit is crucial to the performance of the model. This confirms our intuition that without identifying correct descriptions, the summaries will suffer quality loss. This is simply because out of thousands of descriptions generated by the captioning network, less than 5 percent are selected to be put in the final text synopsis.

6.1.2.3 Purport network

So far, our model is able to 1) generate a sentence, and 2) assign a correctness score to it for every shot in the video. In other words, in the pool of sentences generated for the entire video, our model is able to identify the informative sentences. The final step in the pipeline is to make a coherent text synopsis for the video by choosing the sentences with highest summary-level impact.

To achieve this, we propose to learn it directly from user summaries. Hence, given the visual-language features of all shots, $\{\boldsymbol{f}_1^{vl}, \cdots, \boldsymbol{f}_N^{vl}\}$, we pass them through a bidirectional LSTM to obtain get their forward and backward representations, followed by a fully connected layer with sigmoid nonlinearity to obtain a shot-level significance score β_p.

Using a binary-crossentropy loss, we can learn Purport network's parameters:

$$\mathcal{L}^{\varphi} = -\sum_{p=1}^{N} \varphi_p \log \beta_p + (1 - \varphi_p) \log(1 - \beta_p), \tag{6.3}$$

where φ_p is set to 1 for the shots whose corresponding groundtruth sentences are present in the user summary and it is set to 0 otherwise. It is worth noting that while the purport network is used to essentially summarize the pool of sentences by selecting informative ones, it is **not** a text summarization module. Text summarization models assume that the given document's constituent sentences are correct. However, as we study in section 5.3, a majority of the generated sentences (at least for long egocentric videos) are wrong in describing their corresponding shots. Hence, the purport network uses the visual-language features produced by VLCMU that is responsible in identifying such faults in the generated sentences.

6.1.2.4 Training and inference

To train the model presented in Figure 6.1, first we pre-train the caption generation network. Once it is trained, we freeze its weights to train the rest of the network. The overall objective function of the full pipeline is as the following:

$$\mathcal{L} = \sum_{v=1}^{V} \mathcal{L}_v^c + \lambda_1 \mathcal{L}_v^\eta + \lambda_2 \mathcal{L}_v^\varphi, \tag{6.4}$$

where V is the number of training videos and λ_1 and λ_2 are hyper-parameters that adjust the weights for each term in the overall objective function. At the test time, model generates a sentence for each shot in the video and assigns a correctness and a significance score (α_p and β_p respectively) to each sentence. Summary-level impact value γ_p of a shot is considered as the product of its correctness and significance. These impact values in the temporal order form a time series, where the peaks correspond to locally (in time) important events in the video. While fairly simple and intuitive, there are two main advantages to using such test time inference algorithm. Firstly, since the peaks are local maxima (as opposed to simply choosing sentences with highest γ values), the produced textual synopsis is uniform in time. Secondly, by repeating the inference process multiple times on the γ time-series, one can produce shorter and shorter textual summaries. Each time, we make a smaller time series by only keeping the peaks in previous one. It is easy to infer that text summary is guaranteed to get truncated with a minimum rate of $\frac{1}{2}$ each time the inference algorithm is applied (in worst case, every other point in the time series is a peak, and hence the algorithm discards the non-peak point).

Moreover, after the text synopsis is generated, we can optionally retrieve their corresponding video shots, put them together in temporal order to create a visual synopsis, simulating the standard video summarization functionality.

6.2 Experimental Setup and Results

6.2.1 Dataset

Several video summarization datasets [94, 24, 19] exist, however, most are not readily adaptable for textual video summarization. Furthermore, many consist of short videos (up to 10 minutes long). Video summarization is in fact most helpful when the videos are long and carry high degree of redundancy. Hence, following [25, 26], we train and test our framework (and baselines) on long egocentric videos that Lee et al [33] collected. This dataset consists of 4 videos, each between 3$\sim$5 hours long, covering a variety of daily life activities such as driving, shopping, dining, studying, etc. in uncontrolled environments. Yeung et al. [2] uniformly partitioned each of these videos into non-overlapping 5-second long shots and collected a single sentence description for each via Amazon Mechanical Turk. These descriptions serve as groundtruth in training our caption generation network, and also to infer the pseudo groundtruth labels $\bar{\eta}_p$ used in training our VLCMU.

For any shot index p that is specified to be important φ_p is 1, otherwise it is set to 0. This allows us to learn our purport network parameters via Equation 6.3.

6.2.2 Train-Validation Splits

Given that there are four videos in the UT Egocentric dataset [33], we run four rounds of experiment. Each time, we leave one of the videos out for testing and use the remaining three in the training phase. The pool of shots in the training videos are split again to 80 and 20 percent for training and validation respectively to train the caption generation module. After that, all three (training) videos are used to train the remaining parts of the network. The trained model is then applied on the test video to obtain its text synopsis.

6.2.3 Features

We follow the aforementioned scheme to partition the videos into shots, and subsampled 6 frames from each shot. We use a pretrained GoogLeNet [106] network for the 2D CNN block in Figure 6.1 to extract a 1024-d visual feature representation for each sampled frame. These representations serve as our shot-level features $\{\boldsymbol{f}_p^1, \cdots, \boldsymbol{f}_p^6\}$.

6.2.4 Technical Details

During the training phase, all training sentences are split into their constituent words. Next, we eliminate the words that appear less than four times from the vocabulary. This leaves us with a vocabulary that consists of 461 words in total (this includes start, end, and unknown words). Each of the forward and backward LSTM units in the encoder BLSTM (in the caption generation network) has an output dimension of 512 (this forces the decoder LSTM to output a 1024-d representation). Next a fully connected layer with softmax activation (with output dimension matching the vocabulary size) classifies these representations as words to generate a sentence for each shot. All BLSTM units in the VLCMU and purport network have an output dimension of 128. Thus, each $\boldsymbol{f}_p^{vl}$ is 512-d. We set hyper-parameters λ_1 and λ_2 of Equation 6.4 to 1, hence, giving equal weights to each term in the total objective function.

6.2.5 Details of Inference Algorithm

Our inference algorithm is designed such that the user is able to obtain summaries of different granularity. Therefore, each time we run the inference algorithm to further refine the summary for a less detailed summary. In other words, the peaks that were selected at the previous run, are fed as the new input to the inference algorithm. In our experiments, we run the inference algorithm 4 times for all videos. This results in summaries that are 5 percent (or less) of the overall length of the videos (i.e., 95 percent of sentences/shots are discarded).

Table 6.1: Comparison results between our proposed approached and several state-of-the-art video summarization algorithms that are adapted to produce textual synopsis. All numbers reported below are F1 scores reported by the metric used.

	[1]	[24]	[28]	[81]	[4]	[36]	**Ours**
ROUGESU4	11.95	15.59	16.19	14.10	16.24	13.98	**17.33**
ROUGE-L	11.41	13.88	16.24	13.95	16.99	12.84	**24.98**
METEOR	19.02	18.36	18.90	18.02	**19.90**	17.32	19.27
BLEU2	16.69	34.68	35.28	35.38	35.11	31.72	**46.90**

6.2.6 Evaluating Text Synopses

Since existing video summarization frameworks work in the video domain (i.e. they produce a visual synopsis as opposed to a textual summary), they are not readily comparable to our model. To overcome this problem, we propose to accompany these frameworks with a video caption generation network. In other words, we adapt them to produce textual synopsis for a video. Given a video, each baseline summarizes the video by selecting its key shots. These shots are then passed one by one through caption generation network, identical to that employed in our model (bottom left corner of Figure 6.1), that produces a sentence describing each shot. All sentences of selected key shots are pooled together to construct the textual summary. This resembles how Sah et al. [82] tackled the problem of text synopsis generation for videos. In this work, we use more complex summarization methods that outperform their model. Now that the baselines are adapted to produce textual synopsis for videos, we can compare all the models using existing NLP-based metrics. Yeung et al. [2] were the first to evaluate video summaries through text. More specifically, they used ROUGE-SU, one of the multiple metrics offered by the ROUGE [3] evaluation toolbox. To ensure a more comprehensive study, we also report results using ROUGE-L, METEOR [109], and BLEU2 [60], common metrics in evaluation of machine translation.

Table 6.2: We compare the visual synopsis using AUC metric introduced by [4].

	[1]	[24]	[28]	[81]	[4]	[36]	**Ours**
AUC	12.91	9.85	11.53	12.26	13.14	7.57	**13.19**

6.2.7 Evaluating Visual Synopses

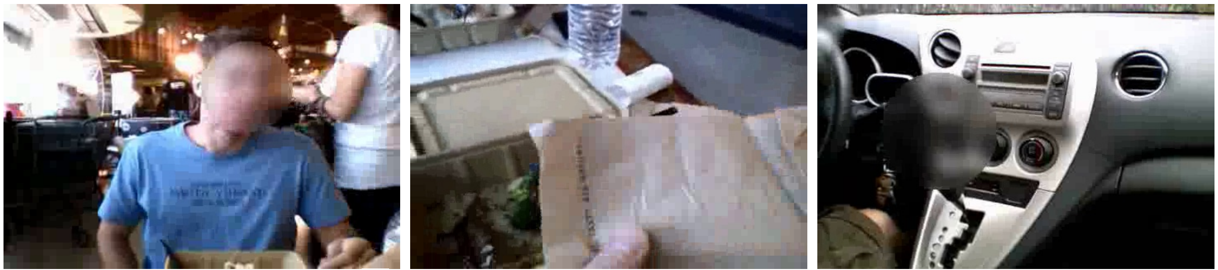
As mentioned earlier, after a text synopsis is generated for the video, we can retrieve their corresponding video shots, put them together in temporal order to create a visual synopsis. The visual synopses are directly comparable to summaries generated by a conventional video summarization method. Therefore, we compare our models against state-of-the-art methods in the visual domain. To do so, we follow [4] by using dense concept annotations to compare the system summaries with their corresponding reference summaries. Given a system summary and a ground truth (user) summary, each shot in the system summary is paired with all the shots in the user summary that are located within a specified temporal vicinity. For each such pair, similarity is defined as intersection-over-union of the pair's concept annotations (concept annotations are binary, i.e., a concept either appears in a shot or not). A maximum weight bipartite graph matching is employed, resulting in an F1-like metric. Finally, the temporal vicinity radius is varied (from 0 to 60 seconds) and area-under-the-curve is calculated.

i looked at the cashier. i looked at the cell. i looked around the restaurant. my friend and i ate frozen yogurt. my friend and i sat at the table. ... . I walked down the street on the sidewalk. i walked down the street. i walked through the building. i walked through the parking lot. ... . i looked at my cell phone. ... . my friend and i sat at the table. ... . i walked through the store. ... . my friend drove the car , and i sat in the passenger seat. ... . i looked at the products. ... . i walked down the street on the sidewalk. i walked through the parking lot. ... . i walked through the store. i washed the dishes. i looked at the laptop. ... . i washed the dishes. ... . i looked at the tablet. ... . i added the ingredients in the pot. i washed the dishes. ... . i looked at the window.

Figure 6.2: Generated textual summary for video 1 in the UT Egocentric dataset is illustrated below. The original summary consists of 144 sentences of which we only show a few here.

6.2.8 Evaluating Caption Generation Network

The first step to generate textual summaries for a given video is to have a video caption generation network that produces descriptions for each shot in the video. Instead of using an off-the-shelf method, we designed our own model, illustrated in the bottom left corner of Figure 6.1. In this section, we study its performance in isolation.



My friend and I sat at the table.✓
My friend and I sat at the table and ate a meal together.

I picked up the baking paper.✗
I used a napkin.

I set the the <UNK>.✗
I sat in the passenger seat.

Figure 6.3: Success and failure cases in the sentences generated by our caption generation network. Generated sentences are shown in blue whereas black color shows the groundtruth descriptions. The sentence generated for the shot on the left (represented by only one of its many frames) is informative. However, for the shot in the middle the generated sentence is wrong in describing what is happening in the shot. And for the shot on the right, the generated sentence is simply meaningless. Hence, it is crucial to train a discriminator such as our VLCMU to distinguish the informative sentences from the rest.

Yeung et al. [2] uniformly partitioned each video in UT Egocentric dataset into non-overlapping 5-sec long shots and collected a textual description for each. Almost all events in the videos are longer than shot length. Splitting the events into short shots results in multiple visually similar clips that have the same or similar descriptions, which allows us to effectively train our caption generation network (we need more than one sample per description to train a network that can generalize). It is worth mentioning that even if the captioning network produces identical descriptions for consecutive shots, this does not reduce the performance. This is due to the fact that we remove duplicate sentences that appear consecutively during inference, hence, all events (whether long or short) are described in a single short sentence. Hence, uniformly splitting the videos does not pose a problem.

To evaluate our video caption generation network, we select one out of four videos in the dataset (UT Egocentric) for testing, and use the remaining three to train the network. We divide the videos into shots and use each shot and its corresponding groundtruth sentence to train the models. 80 percent of training {shot,description} pairs are used for training whereas the remaining 20 percent are used for validation. We train the network for 20 epochs with batch size of 64 using "RMSPROP" optimization algorithm and validate it after every epoch on the validation set. Finally, we load the model that performs the best on the validation data and use it to generate sentences for the test video shots. Since there are four videos in the UT Egocentric dataset, each time we leave one out for testing and train the model from scratch on the remaining videos and report the performance using the ROUGE-SU4 F1 score in table 6.3 below.

Table 6.3: Per-video comparison (ROUGE-SU4 F1 score) of our video caption generation network.

Video 1	Video 2	Video 3	Video 4	Average
20.52	11.09	17.40	22.54	17.89

As we can observe from the table 6.3 the ROUGE-SU4 scores are fairly low. Evidently, the majority of the sentences generated are uninformative. Some of these sentences are meaningless (Figure 6.3 on the right), whereas some others are simply wrong in describing what actually is happening in the video shot (Figure 6.3 in the middle). This is why we proposed the VLCMU in order to identify the informative sentences.

6.2.9 Results Analysis

As mentioned earlier, for each video in the dataset, three groundtruth text summaries are available. Each system summary is compared to all three reference summaries. Therefore, we obtain three F1 scores (using the precision and recall reported by ROUGE-SU4) for each system generated summary and report their average. We compare our model with several video summarization methods listed above in Table 6.1. We are able to achieve state-of-the-art performance on every video in the UT Egocentric dataset. As we can observe from the table, SubMod [28] and LM-SeqGDPP [4] have relatively higher performance compared to the other approaches. This is in fact because both of these methods are fed the expected summary length. In other words, they produce system summaries with exact number of sentences that of in their corresponding groundtruth summaries. Since F1 score is the harmonic mean of precision and recall, it leads to an advantage for these methods. It is also worth noting that even though LSTMDPP [36] also has this advantage, it is unable to perform well. This is because the summaries produced by this approach are *not* uniform in time and repetition can be observed in their summaries. SeqDPP by [1] produced long summaries compared to other methods as it has no mechanism for controlling the summary length. This leads to its fairly low performance in our experiments. In the case of Video 3 and 4, the summaries produced by our method were roughly 50 percent shorter than the groundtruth summaries, and even so, we outperform all other baselines. Only the summary generated for Video 1 was slightly longer than its groundtruth. Figure 6.2 illustrates a sample summary from our model.

We also evaluated **visual** synopses under the AUC metric of [4]. As shown in Table 6.2, we outperform the existing state-of-the-art. This shows that transferring the summarization task from the visual domain to text domain results in generating better summaries in **both** domains.

6.2.10 Ablation Study

To illustrate the effectiveness of the introduced VLCMU and purport network, we remove each from the pipeline with minimal changes to the remaining structure. As we explained earlier, the purpose of our VLCMU is to identify which generated captions are informative. Hence, we expect to observe a noticeable drop in performance when we eliminate this module. When this module is entirely removed (i.e., including the BLSTM units that process visual features and generated text), we notice a performance drop of 1.43 percent in the F1 score. However, we can keep the BLSTM units and only remove the auxiliary loss $\mathcal{L}^{\eta}$ (and the corresponding fully connected layer that produces α_p that is designed to enhance learning its parameters. When $\mathcal{L}^{\eta}$ is removed from the objective function, we observe an insignificant drop in performance. This is in fact because we use pseudo-groundtruth labels to calculate $\mathcal{L}^{\eta}$ to avoid extra annotation. In another experiment, we remove the purport network. In this experiment, the pseudo groundtruth labels $\bar{\eta}_p$ in $\mathcal{L}^{\eta}$ are replaced with φ_p. In this case, we observe that the performance drops 0.81 percent in F1 score. This is expected as we are only considering each generated on its own and independent of all other sentences.

Table 6.4: Ablation study. We study the effect of removing components, specified by column title, from the pipeline.

	-VLCMU	-$\mathcal{L}^{\eta}$	-Purport	[110]
R-SU4	15.90	17.12	16.52	12.41

In addition, we design an experiment in which we remove both the VLCMU and purport network from the pipeline, and instead apply a document summarization method to obtain the textual summary. To do this, we generate the descriptions for all shots in a video and put them together in a text document. This document is then summarized using the state-of-the-art extractive document summarization method of [110]. This yields a low performance of 12.41 under ROUGE-SU4 F1-score. This is expected as such approaches do not account for the fact that some of the generated descriptions may be incorrect. We summarize these observations in Table 6.4.

6.3 Summary

In this Section, we presented a framework that given a video, it produces a short textual synopsis for it. To this end, the video is divided into shots and a descriptive sentence for each shot is generated via a video caption generation network. Since some of the generated sentences may be uninformative specially when dealing with long egocentric videos, we developed a Visual-Language Content Matching Unit that can distinguish between the informative and uninformative sentences. Next, our purport network reads the generated sentences in temporal order, to select those that convey the most information about the video. Our framework can also generate a visual synopsis for the video by retrieving the shots which their corresponding descriptions were included in the textual synopsis and stitching them together in temporal order to make a short video. We evaluated our model on a challenging egocentric dataset where videos are over 3 hours long, and achieve state-of-the-art performance.

CHAPTER 7: CONCLUSION AND FUTURE WORK

In this Chapter, we highlight the concluding remarks on this dissertation, and expand on potential future work in this direction of research.

7.1 Conclusion

In this Dissertation, we study the problem of video summarization. Our aim to tackle the inherent subjectivity present in this area of research by presenting frameworks that allow us to personalize the summarization process for the specific user given their preferences. To this end, we put together a dataset, alongside with annotations required to train the models. Moreover, we design a novel evaluation metric upon the collected annotations to enhance the automatic evaluation of system generated summaries. Furthermore, to train better summarization models, we re-design the training objective function to address the inconsistencies between the training and testing schemes. Lastly, to facilitate faster information extraction from the video summaries, we develop an architecture that produces a short textual summary for a given video.

In Chapter 3, we examine a query-focused video summarization problem, in which the decision to select a video shot to the summary depends on both 1) the relevance between the shot and the query and 2) the importance of the shot in the context of the video. To tackle this problem, we developed a probabilistic model, Sequential and Hierarchical Determinantal Point Process (SH-DPP), as well as efficient learning and inference algorithms for it. Our SH-DPP summarizer can conveniently handle extremely long videos or online streaming videos. On two benchmark datasets for video summarization, our approach significantly outperforms some competing baselines. To the best of our knowledge, ours is the first work on query-focused video summarization, and has a great potential to be used in search engines, e.g., to display snippets of videos.

In Chapter 4, we study the *subjectiveness* in video summarization. On our course to find a solution, we compiled a dataset that is densely annotated with a comprehensive set of concepts and designed a novel evaluation metric that benefits from the collected annotations. We also devised a new approach to generating personalized summaries by taking user queries into account. We employed memory networks and determinantal point processes in our summarizer, so that our model leverages their attention schemes and diversity modeling capabilities, respectively.

In Chapter 5, 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, *G*DPP, 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 hours. We collected 3 user-annotated summaries per video as well as dense concept annotations required for the evaluation.

In Chapter 6, we present a framework that given a video, it produces a short textual synopsis for it. To this end, the video is divided into shots and a descriptive sentence for each shot is generated via a video caption generation network. Since some of the generated sentences may be uninformative specially when dealing with long egocentric videos, we develop a visual-language content matching unit that can distinguish between the informative and uninformative sentences. Next, our purport network reads the generated sentences in temporal order, to select those that convey the most information about the video. To the best of our knowledge, we are the first to develop a unified approach to generating text synopsis for videos by modeling the problem as dense text generation followed by text summarization under severe noise. Our framework can also generate a visual synopsis for the video by retrieving the shots whose corresponding descriptions were included in the textual synopsis and stitch them together in temporal order to make a short video.

7.2 Future Work

Here we highlight some potential extensions and areas of future research that could be explored to further study the visual-textual summarization of videos.

7.2.1 Assumptions and Shortcomings

When training a video summarization model in a supervised learning scenario, an oracle summary is used as the groundtruth as reference to train model on. As mentioned throughout this Dissertation, video summarization is inherently subjective and hence, defining a gold standard summary is a difficult task. To deal with this issue, we ask different users to summarize the same video (as well as the same query in the case of query-focused summarization), and then we merge such summaries using a greedy approach. This technique compares every shot in the video with shots selected by different users, and at every iteration adds the shots that is most similar to those selected by the users. In other words, by using this technique, we convert the user-specified summaries into an "average user" summary. This does not pose a problem in the case of generic video summarization, however, for the purpose of query-focused summarization, this approach can be limiting. Ideally, a query-focused video summarization model should learn directly from the user summaries (and not the average user summary).

Another limitation of our query-focused summarization models is the restricted concept dictionary used to define queries. Ideally, the query should be of free-form language nature. Although, we do not explore such queries in this Dissertation, theoretically, our approach in Chapter 4 can be extended to accept such queries with minimal changes to its structure.

The models introduced in Chapters 3 and 4 do not have a mechanism to allow user input on the expected summary length. Therefore the system determines the length of the summary on its own. Therefore, at the evaluation step, when we compare the system summary with the groundtruth summary, we have summaries of different lengths. This difference in lengths leads to less accurate comparison of the summaries. The reason is that when matching shots between summaries, some shots may remain unmatched or there could be multiple matches for some shots.

In Chapter 6, we generate text summaries for the videos. In addition, we could create the visual summary by temporally concatenating the shots corresponding to the sentences in the text summary. One issue that manifests is that although the sentence might be descriptive of the shot, the clip itself might not be the most representative shot that we could select for the description. One possible solution to this issue, is to rather search within the vicinity for the most representative shot.

7.2.2 Alternative approaches, problems, and setups

The problem of query-focused video summarization can be modeled in various ways. In this Dissertation, we expect the user to explicitly define the query for the system. This is useful in the case of search engines. However, if the goal is to have an automated mean to create personalized summaries for the user, it is best if the method is able to infer user's interests. Such systems may require feedback from the user to identify the user's preferences, and directly apply those when summarizing a video.

To extend query-focused video summarization framework to accept queries of free-form language type, one can apply object detectors on video shots, and use distance between the words in the query and the objects found in the shots to summarize the videos. Another advantage to such model is its flexibility to extend to new words in the query with minimal effort.

Another way to benefit from text in video summarization, is to use text features to enrich the visual features. Video summarization is a complex task and can benefit from a high-level understanding to enable better summarization of a video. The visual-language features can help to, 1) create a better visual synopsis, and 2) generate a textual summary for the video. Such enriched features can be learned via a two-stream network that either tries to match the distributions of visual and textual features, or learn an embedding where matching visual features and text descriptions are close to one another.

LIST OF REFERENCES

[1] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, “Diverse sequential subset selection for supervised video summarization,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2069–2077.

[2] S. Yeung, A. Fathi, and L. Fei-Fei, “Videoset: Video summary evaluation through text,” *arXiv preprint arXiv:1406.5824*, 2014.

[3] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8, 2004.

[4] A. Sharghi, A. Borji, C. Li, T. Yang, and B. Gong, “Improving sequential determinantal point processes for supervised video summarization,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 517–533.

[5] R. M. Jiang, A. H. Sadka, and D. Crookes, “Advances in video summarization and skimming,” in *Recent Advances in Multimedia Signal Processing and Communications*. Springer, 2009, pp. 27–50.

[6] A. Rav-Acha, Y. Pritch, and S. Peleg, “Making a long video short: Dynamic video synopsis,” in *CVPR, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006.

[7] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz, “Schematic storyboarding for video visualization and editing,” in *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 862–871.

[8] T. Liu and J. R. Kender, “Optimization algorithms for the selection of key frame sequences of variable length,” in *Computer VisionECCV 2002*. Springer, 2002, pp. 403–417.

[9] A. Aner and J. R. Kender, "Video summaries through mosaic-based shot and scene clustering," in *Computer VisionECCV 2002*. Springer, 2002.

[10] N. Vasconcelos and A. Lippman, "A spatiotemporal motion model for video summarization," in *CVPR, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 361–366.

[11] W. Wolf, "Key frame selection by motion analysis," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 1228–1231.

[12] K. M. Lee and J. Kwon, "A unified framework for event summarization and rare event detection," in *2012 IEEE Conference on CVPR*. IEEE, 2012.

[13] Y. Cong, J. Yuan, and J. Luo, "Towards scalable summarization of consumer videos via sparse dictionary selection," *Multimedia, IEEE Transactions on*, vol. 14, no. 1, pp. 66–75, 2012.

[14] C. Ngo, Y. Ma, and H. Zhang, "Automatic video summarization by graph modeling," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003.

[15] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan, "Large-scale video summarization using web-image priors," in *Proceedings of the IEEE Conference on CVPR*, 2013.

[16] G. Kim, L. Sigal, and E. Xing, "Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in *Proceedings of the IEEE Conference on CVPR*, 2014.

[17] B. Xiong and K. Grauman, "Detecting snap points in egocentric video with a web photo prior," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 282–298.

[18] W.-S. Chu, Y. Song, and A. Jaimes, “Video co-summarization: Video summarization by visual co-occurrence,” in *Proceedings of the IEEE Conference on CVPR*, 2015.

[19] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, “Tvsum: Summarizing web videos using titles,” in *Proceedings of the IEEE Conference on CVPR*, 2015.

[20] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo, “Multi-task deep visual-semantic embedding for video thumbnail selection,” in *Proceedings of the IEEE Conference on CVPR*, 2015.

[21] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, “Category-specific video summarization,” in *Computer Vision–ECCV 2014*. Springer, 2014.

[22] M. Sun, A. Farhadi, and S. Seitz, “Ranking domain-specific highlights by analyzing edited videos,” in *Computer Vision–ECCV 2014*. Springer, 2014.

[23] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, “Gaze-enabled egocentric video summarization via constrained submodular maximization,” in *Proceedings of the IEEE Conference on CVPR*, 2015.

[24] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating summaries from user videos,” in *Computer Vision–ECCV 2014*. Springer, 2014.

[25] Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” in *Proceedings of the IEEE Conference on CVPR*, 2013.

[26] Y. J. Lee and K. Grauman, “Predicting important objects for egocentric video summarization,” *International Journal of Computer Vision*, vol. 114, no. 1, pp. 38–55, 2015.

[27] D. Liu, G. Hua, and T. Chen, “A hierarchical visual model for video object summarization,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 12, pp. 2178–2190, 2010.

[28] M. Gygli, H. Grabner, and L. Van Gool, “Video summarization by learning submodular mixtures of objectives,” in *Proceedings of the IEEE Conference on CVPR*, 2015.

[29] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in *Mining Text Data*. Springer, 2012, pp. 43–76.

[30] A. Kulesza and B. Taskar, “Determinantal point processes for machine learning,” *arXiv preprint arXiv:1207.6083*, 2012.

[31] J. Ghosh, Y. J. Lee, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” in *2012 IEEE Conference on CVPR*. IEEE, 2012.

[32] S. Yeung, A. Fathi, and L. Fei-Fei, “Videoset: Video summary evaluation through text,” *arXiv preprint arXiv:1406.5824*, 2014.

[33] Y. J. Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization.” in *CVPR*, vol. 2, no. 6, 2012, p. 7.

[34] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Summary transfer: Exemplar-based subset selection for video summarizatio,” *arXiv preprint arXiv:1603.03369*, 2016.

[35] B. Zhao and E. Xing, “Quasi real-time summarization for consumer videos,” in *Proceedings of the IEEE Conference on CVPR*, 2014.

[36] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, “Video summarization with long short-term memory,” *arXiv preprint arXiv:1605.08110*, 2016.

[37] S. Sukhbaatar, J. Weston, R. Fergus *et al.*, “End-to-end memory networks,” in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[38] A. Sharghi, B. Gong, and M. Shah, “Query-focused extractive video summarization,” in *European Conference on Computer Vision*. Springer, 2016, pp. 3–19.

[39] A. Sharghi, J. S. Laurel, and B. Gong, “Query-focused video summarization: Dataset, evaluation, and a memory network based approach,” *arXiv preprint arXiv:1707.04960*, 2017.

[40] I. Sutskever, J. Martens, and G. E. Hinton, “Generating text with recurrent neural networks,” in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1017–1024.

[41] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[42] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[43] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, “Grammar as a foreign language,” in *Advances in Neural Information Processing Systems*, 2015, pp. 2773–2781.

[44] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, “Building end-to-end dialogue systems using generative hierarchical neural network models.” in *AAAI*, 2016, pp. 3776–3784.

[45] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” *arXiv preprint arXiv:1511.06732*, 2015.

[46] S. Wiseman and A. M. Rush, “Sequence-to-sequence learning as beam-search optimization,” *arXiv preprint arXiv:1606.02960*, 2016.

[47] A. Kulesza and B. Taskar, “k-dpps: Fixed-size determinantal point processes,” in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 1193–1200.

[48] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1226–1233.

[49] M. Ellouze, N. Boujemaa, and A. M. Alimi, "Im (s) 2: Interactive movie summarization system," *Journal of Visual Communication and Image Representation*, vol. 21, no. 4, pp. 283–294, 2010.

[50] B. Xiong, G. Kim, and L. Sigal, "Storyline representation of egocentric videos with an applications to story-based search," in *Proceedings of the IEEE International CVPR*, 2015.

[51] H. Daumé III and D. Marcu, "Bayesian query-focused summarization," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006.

[52] F. Schilder and R. Kondadadi, "Fastsum: fast and accurate query-based multi-document summarization," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008, pp. 205–208.

[53] S. Gupta, A. Nenkova, and D. Jurafsky, "Measuring importance and query relevance in topic-focused multi-document summarization," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, 2007, pp. 193–196.

[54] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, "Towards ai-complete question answering: A set of prerequisite toy tasks," *arXiv preprint arXiv:1502.05698*, 2015.

[55] J. Weston, S. Chopra, and A. Bordes, “Memory networks,” *arXiv preprint arXiv:1410.3916*, 2014.

[56] C. Xiong, S. Merity, and R. Socher, “Dynamic memory networks for visual and textual question answering,” *arXiv preprint arXiv:1603.01417*, 2016.

[57] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433.

[58] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence-video to text,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.

[59] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[60] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[61] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, “Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2712–2719.

[62] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, “Translating video content to natural language descriptions,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 433–440.

[63] S. Chen, J. Chen, and Q. Jin, "Generating video descriptions with topic guidance," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017, pp. 5–13.

[64] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, "Semantic compositional networks for visual captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5630–5639.

[65] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *International Conference on Machine Learning*, 2014, pp. 595–603.

[66] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4594–4602.

[67] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 706–715.

[68] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu, "Bidirectional attentive fusion with context gating for dense video captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7190–7198.

[69] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8739–8748.

[70] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, "Coherent multi-sentence video description with variable level of detail," in *German conference on pattern recognition*. Springer, 2014, pp. 184–195.

[71] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, “Video paragraph captioning using hierarchical recurrent neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4584–4593.

[72] D. R. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization,” *Computational linguistics*, vol. 28, no. 4, pp. 399–408, 2002.

[73] J. G. Carbonell and J. Goldstein, “The use of mmr and diversity-based reranking for reodering documents and producing summaries,” 1998.

[74] J. Cheng and M. Lapata, “Neural summarization by extracting sentences and words,” *arXiv preprint arXiv:1603.07252*, 2016.

[75] H. P. Edmundson, “New methods in automatic extracting,” *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.

[76] R. McDonald, “A study of global inference algorithms in multi-document summarization,” in *European Conference on Information Retrieval.* Springer, 2007, pp. 557–564.

[77] S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93–98.

[78] D. Pighin, M. Cornolti, E. Alfonseca, and K. Filippova, “Modelling events through memory-based, open-ie patterns for abstractive summarization,” 2014.

[79] K. Thadani and K. McKeown, “Supervised sentence fusion with single-stage inference,” in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 1410–1418.

[80] K. Woodsend and M. Lapata, “Multiple aspect summarization using integer linear programming,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 233–243.

[81] B. A. Plummer, M. Brown, and S. Lazebnik, “Enhancing video summarization via vision-language embedding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5781–5789.

[82] S. Sah, S. Kulhare, A. Gray, S. Venugopalan, E. Prud’Hommeaux, and R. Ptucha, “Semantic text summarization of long videos,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 989–997.

[83] J. B. Hough, M. Krishnapur, Y. Peres, B. Virág *et al.*, “Determinantal processes and independence,” *Probability surveys*, vol. 3, pp. 206–229, 2006.

[84] A. Borodin and E. M. Rains, “Eynard–mehta theorem, schur process, and their pfaffian analogs,” *Journal of statistical physics*, vol. 121, no. 3, pp. 291–317, 2005.

[85] A. Kulesza and B. Taskar, “Learning determinantal point processes,” *arXiv preprint arXiv:1202.3738*, 2012.

[86] W.-L. Chao, B. Gong, K. Grauman, and F. Sha, “Large-margin determinantal point processes.”

[87] R. H. Affandi, A. Kulesza, and E. B. Fox, “Markov determinantal point processes,” *arXiv preprint arXiv:1210.4850*, 2012.

[88] B. Gong, W. Chao, K. Grauman, and F. Sha, “Diverse sequential subset selection for supervised video summarization,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2069–2077.

[89] D. Borth, T. Chen, R. Ji, and S.-F. Chang, “Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content,” in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013.

[90] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[91] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[92] F. Yu, L. Cao, R. Feris, J. Smith, and S.-F. Chang, “Designing category-level attributes for discriminative visual recognition,” in *Proceedings of the IEEE Conference on CVPR*, 2013.

[93] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg, “From large scale image categorization to entry-level categories,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2768–2775.

[94] S. E. F. De Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Araújo, “Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method,” *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[95] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” *arXiv preprint arXiv:1504.00325*, 2015.

[96] H. Daumé, J. Langford, and D. Marcu, “Search-based structured prediction,” *Machine learning*, vol. 75, no. 3, pp. 297–325, 2009.

[97] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[98] M. Collins and B. Roark, "Incremental parsing with the perceptron algorithm," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 111.

[99] S. Shen, Y. Cheng, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Minimum risk training for neural machine translation," *arXiv preprint arXiv:1512.02433*, 2015.

[100] H. Daumé III and D. Marcu, "Learning as search optimization: Approximate large margin methods for structured prediction," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 169–176.

[101] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," *Foundations and Trends® in Machine Learning*, vol. 5, no. 2–3, pp. 123–286, 2012.

[102] D. J. Aldous, "Some inequalities for reversible markov chains," *Journal of the London Mathematical Society*, pp. 564–576, 1982.

[103] R. Bubley and M. Dyer, "Path coupling: A technique for proving rapid mixing in markov chains," in *focs*, 1997, pp. 223–231.

[104] M. Dyer and C. Greenhill, "A more rapidly mixing markov chain for graph colorings," *Random Structures and Algorithms*, pp. 285–317, 1998.

[105] C. Li, S. Jegelka, and S. Sra, "Fast dpp sampling for nystr\" om with application to kernel methods," *arXiv preprint arXiv:1603.06052*, 2016.

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

[107] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang, "Large-scale visual sentiment ontology and detectors using adjective noun pairs," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 223–232.

[108] X. Duan, W. Huang, C. Gan, J. Wang, W. Zhu, and J. Huang, "Weakly supervised dense event captioning in videos," in *Advances in Neural Information Processing Systems*, 2018, pp. 3059–3069.

[109] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[110] Y. Dong, Y. Shen, E. Crawford, H. van Hoof, and J. C. K. Cheung, "Banditsum: Extractive summarization as a contextual bandit," *arXiv preprint arXiv:1809.09672*, 2018.