"*Previously on ...*" From Recaps to Story Summarization

Aditya Kumar Singh Dhruv Srivastava Makarand Tapaswi
CVIT, IIIT Hyderabad, India

https://katha-ai.github.io/projects/recap-story-summ/

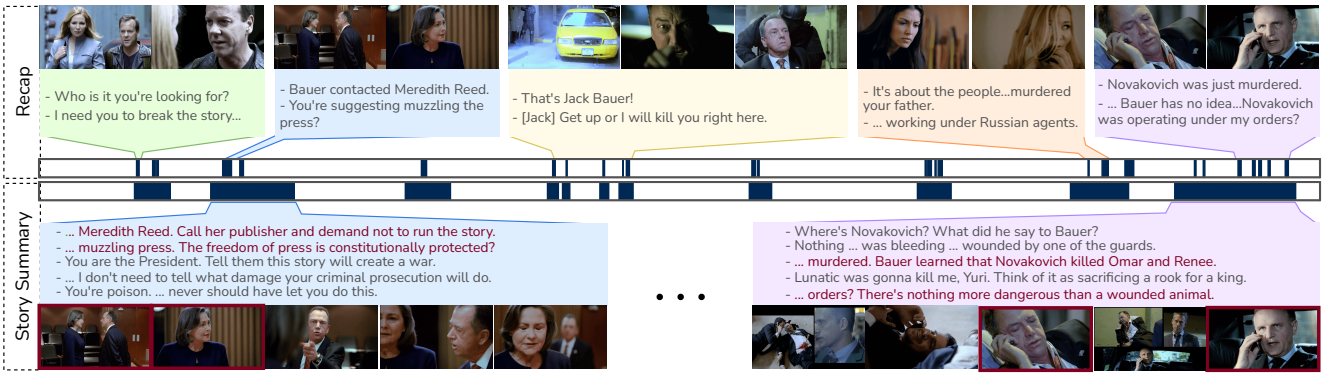

Figure 1. We illustrate how TV show recaps can be used to generate labels for *multimodal story summarization*. The *top half* features the recap shown at the beginning of the episode S08E23 based on key moments (shots and dialogs) from S08E22 of the TV series *24*. As recaps help viewers recall essential story events, we extend these aligned segments to create summarization labels (visualized in the *bottom half* where the actual shots and dialogs inherited from recap are marked in deep red). For example, in the sub-story (left), the recap hints at Jack Bauer relaying classified information to the press, while the summary presents the complete sub-story, including Logan informing President Taylor about their failure to catch Jack and their disagreement over muzzling the press.

Abstract

We introduce multimodal story summarization *by leveraging TV episode recaps – short video sequences interweaving key story moments from previous episodes to bring viewers up to speed. We propose* PlotSnap*, a dataset featuring two crime thriller TV shows with rich recaps and long episodes of 40 minutes. Story summarization labels are unlocked by matching recap shots to corresponding sub-stories in the episode. We propose a hierarchical model* TaleSumm *that processes entire episodes by creating compact shot and dialog representations, and predicts importance scores for each video shot and dialog utterance by enabling interactions between local story groups. Unlike traditional summarization, our method extracts multiple plot points from long videos. We present a thorough evaluation on story summarization, including promising cross-series generalization. TaleSumm also shows good results on classic video summarization benchmarks.*

1. Introduction

Imagine settling in to catch the latest episode of our favorite TV series. We hit play and the familiar "*Previously on ...*", the recap, a smartly edited segment swiftly brings us up to speed, reminding us of key moments from past episodes.

A TV show **recap** is a concise, under-two-minute sequence of crucial plot points from previous episodes. To satisfy the time constraint, the recap is constructed by editing shots from previous episode with sharp and rapid cuts and selecting/modifying dialog utterances to ensure relevance to the sub-story. A good recap sets the stage for the main part of the episode by weaving visual and dialog cues to *spark the viewers' memory*. Thus, a recap is a great way to identify sub-stories important to the overall story arc.

We use recaps to create **story summaries** by identifying and expanding the sub-stories from the episode (Fig. 1). We introduce an innovative shot-matching algorithm (Sec. 3) that associates shots from the recap to their corresponding shots in the episode. Different from a recap, a story summary consists of entire scenes or sub-stories that are essential to the narrative. Thus, a first-time viewer may watch story summaries of each episode serially and understand the main narrative, while watching recaps serially does not help as they are only meant as memory triggers and assume that the viewer has seen the episode before.

We propose a novel task of **creating multimodal story**

Dataset	Modalities	#	Length	Content	Summary Annotations
SumMe [21]	V2V	25	1-6 min	Holidays, events, sports	Multiple set of key fragments
TVSum [75]	V2V	50	1-11 min	News, how-to, user-generated, documentary	Multiple fragment-level scores
OVP [11]	V2V	50	1-4 min	Documentary, educational, historical, lecture	Multiple set of key-frames
CNN-DailyMail [51]	T2T	311672	766 words	News articles and highlight stories	Human-generated internet summaries
XSum [53]	T2T	226711	431 words	BBC News articles	Single-sentence summary by author
TRIPOD [56]	T2T	99	22072 words	Movies (action, romance, comedy, drama)	Synopses level annotations
SummScreen [8]	T2T	26851	7013 words	TV screenplays (wide scope, 21 genres)	Human-written internet summaries
How2 [72]	VT2T	79114	2-3 min	Instructional videos	Youtube descriptions (and translations)
SummScreen3D [55]	VT2T	4575	5721 words	TV Shows (soap operas)	Human written internet summaries
BLiSS [25]	VT2VT	13303	10.1 min/49 words	Livestream Videos	Human text-summaries; Thumbnail animation
PlotSnap (Ours)	VT2VT	215	40-45 min	TV Shows (crime thriller)	Matching recap shots followed by smoothing

Table 1. Overview of video/text/multimodal summarization datasets. # indicates the size of the dataset (no. of instances). The modalities column includes: V2V: Video-to-video, T2T: Text-to-text, VT2T: Video-text to text, and VT2VT: Video-text to video-text summarization. Closest to our domain of story summarization are SummScreen and SummScreen3D, however, they produce text summaries.

summaries for TV episodes. We introduce *PlotSnap*, a new dataset for story summarization consisting of two popular crime thrillers: (i) *24* [1] features Jack Bauer, an agent at the counter-terrorism unit who relentlessly tackles seemingly impossible missions; and (ii) *Prison Break* [2] features Michael Scofield who plans and executes daring escapes from prisons. We choose action thrillers as they are often more challenging than romantic and situational comedies with multiple suspenseful story-lines, rapid action sequences, and complex visual scenes. With excellent recaps in both shows, we can extract important narrative subplots from the recap to create story summaries (see Sec. 3).

Our task of story summarization is an instance of multimodal long-video understanding where an entire episode (typically 40 minutes) needs to be processed. We formulate story summarization as an extractive multimodal summarization task with multimodal outputs (video-text to video-text, VT2VT). Specifically, we build models that predict the importance of each video shot and dialog utterance (story elements) in an episode. Selecting multiple major and connected sub-stories is different and challenging from most summarization works that promote visual diversity [39].

We also propose a new hierarchical Transformer model, TaleSumm, to perform story summarization. Different from typical summarization approaches [16, 55, 95] that use multimodal inputs to either generate a video (select frames) *or* a text summary, our model predicts scores for both modalities. Recent multimodal approaches, A2Summ [25] and VideoXum [44], also generates both outputs; but we differ significantly in video type (stories *vs*. creative videos), the duration of the input video, and the model architecture. The first level of our model encodes shot and utterance representations. At the second level, we foster interaction between shots and utterances within local story groups based on a temporal neighborhood, reducing the impact of distant and potentially noisy elements. A dedicated group token enables message-passing across story groups.

In summary, our contributions are: (i) We propose story summarization that requires identifying and extracting multiple plot points from narrative content. This is a challenging multimodal long-video understanding task. (ii) We pioneer the use of TV show recaps for video understanding and show their application in story summarization. We introduce PlotSnap, a new dataset featuring 2 crime thriller TV series with rich recaps. (iii) We propose a novel hierarchical model that features shot and dialog level encoders that feed into an episode-level Transformer. The model operates on the full episode while being lightweight enough to train on consumer GPUs. (iv) We present an extensive evaluation: ablation studies validate our design choices, TaleSumm obtains SoTA on PlotSnap and performs well on video summarization benchmarks. We show generalization across seasons and even across TV shows, and evaluate consistency of labels obtained from multiple diverse sources.

2. Related Work

Video summarization predates Deep Learning (DL). Past methods focused on generating keyframes [36, 37, 40, 86], skims [20, 46], video storyboards [19], time-lapses [38], montages [77], or video synopses [63]. However, given the effectiveness of DL methods (*e.g*. [22, 34, 47, 90]) over traditional optimization-based approaches, we will primarily discuss learning-based approaches in the following.

Summarization modalities. We classify approaches based on input and output modalities. **(i)** Video to frames/video (V2V) approaches model temporal relations [92, 96, 97], preserve diversity [39, 91], or generate images/videos [4, 15, 67, 93]. On the other, **(ii)** text to text (T2T) methods are either *extractive* [33, 45, 94] picking important sentences from a document, or *abstractive* [43, 66, 89] summarizing the overall meaning by generating new text [50]. Relevant to our work, story screenplay summaries [8, 57] or turning point identification [56] can be seen as T2T summarization.

Multimodal approaches typically benefit from additional modalities to enhance model performance. **(iii)** Video-text

to text (VT2T) is popular for screenplays [55, 59], particularly in generating video captions [4, 68, 72]. **(iv)** Video-text to video (VT2V) covers the field of *query-guided summarization* [30, 52, 74]. Finally, the last option is **(v)** video-text to video-text (VT2VT) summarization. Our work lies here and is different from A2Summ [25] and VideoXum [44], as we operate on long videos edited to convey complex stories. Different from trailer generation [58] that avoids spoilers, we wish to identify all key story events.

Summarization datasets. We compare popular summarization datasets based on above modalities in Table 1. Video-only datasets, TVSum [75] and SumME [21], consist of short duration videos unlike ours. Other video datasets work with first-person videos [26], are used for title generation [88], and even feature e-sports audience reactions [14]. For a nice overview of text-only (T2T) and text-primary (VT2T) datasets, we refer the reader to [55]. Briefly, text datasets include news articles (CNN-DailyMail [51], XSum [53]), human dialog (Samsum [18]), and TV/movie screenplays (SummScreen [8]). While similar in spirit to screenplays used for storytelling, PlotSnap is different as it features TV episodes with long videos and dialogs (without speaker labels or scene descriptions), a significant challenge in long-form video understanding.

Story summarization retrieves multiple sub-stories contained within the story-arc of an episode. To our best knowledge, we are unaware of works on video-text story-summary generation. There are attempts to understand stories in movies/TV shows through various dimensions: person identification [24, 48, 49, 83], question-answering [41, 42, 82], captioning [60, 62, 69, 70], situation understanding [35, 71, 85, 87], text alignment [73, 79, 81, 98], or scene detection [9, 23, 31, 32]. Recently, SummScreen3D [55] extends SummScreen [8] with visual inputs, but the output summary is still textual. On the other hand, our goal is multimodal story-summary generation by predicting both - important video shots *and* dialogs.

3. PlotSnap Dataset

We introduce the PlotSnap dataset consisting of long-form multimodal TV episodes with a well-structured underlying plot spanning multiple seasons and episodes. We consider two American crime thriller TV shows with rich storylines: *24* [1] and *Prison Break* (PB) [2]. Unlike sitcoms, crime thrillers are recognized for their methodically crafted captivating plot lines. Notably, both *24* and *Prison Break* have good recaps, and are famous for using the catchphrase "*Previously on ...*" at the start of the recap.

We present some statistics of PlotSnap in Table 2. With a total of 205 episodes, the large number of shots and dialogs present in each episode pose interesting challenges for summarization. The first section of the table presents overall size and duration, second shows statistics for shots and dialog utterances, and the third for recaps. We note that recap shots are much shorter (1.9s *vs*. 3.2s for *24*) allowing the editors to pack more story content in the same duration.

TV Series	24	Prison Break
# of Seasons	8	2
# of Episodes	172	33
Dataset duration (hours)	125.9	24.0
Avg episode duration (s)	2635 ± 72	2615 ± 39
Avg # of shots per episode	825 ± 101	999 ± 117
Avg duration of shots (s)	3.2 ± 2.5	2.6 ± 2.3
Avg # of utterances per episode	564 ± 54	529 ± 59
Avg # of words/tokens in utterance	7.9 ± 5.4	7.4 ± 5.8
Avg recap duration (s)	104 ± 28	62 ± 20
Avg # of shots in recap	55 ± 12	43 ± 9
Avg # of utterances in recap	33 ± 6	22 ± 5

Table 2. Mean ($\pm$ stddev) featuring properties of video shots, dialog utterances, and the recap in our dataset PlotSnap.

Our key idea is to use professionally edited recaps, shown at the beginning of a new episode, as labels for story summarization. Let $\mathcal{E}_n$ be the n^{th} episode in a TV series. $\mathcal{R}_{n+1}$ is the recap shown just before the episode $\mathcal{E}_{n+1}$ begins and may contain content from all past episodes $\{\mathcal{E}_n, \ldots, \mathcal{E}_1\}$. Thus, we classify the visual content appearing in the recap into three sources along with their average proportions (for *24*): (i) shots that are picked (and usually trimmed) from $\mathcal{E}_n$ (88%), (ii) shots that are picked from $\mathcal{E}_{n-1}$ or earlier (5%), and (iii) new shots that did not appear in any previous episode (7%). As most shots (88%) of a recap are from the preceding episode, recaps serve as good summary labels. The remaining 12% recap content (from earlier episodes or unseen shots) is ignored. We also remove the last episode of each season due to the absence of a recap.

Recap inspired labels. We present how recap shots and dialogs can be used to create labels for story summarization. First, we manually extract the recap ($\mathcal{R}_{n+1}$) from $\mathcal{E}_{n+1}$ instead of employing automatic detection methods [23] to avoid introducing additional label noise. Second, to localize trimmed recap shots in past episodes ($\mathcal{E}_1, \ldots, \mathcal{E}_n$), we propose a shot-matching algorithm that conducts pairwise comparisons of frame-level embeddings, making selections based on a threshold determined by similarity score and frequency. Due to *shot thread* patterns [80], one recap shot may match multiple shots in the episode. This is desirable as we want to highlight larger sub-stories as part of the summary. In fact, selecting only one shot in a thread adversely affects the model due to conflicting signals as shots with very similar appearance are assigned opposite labels.

We think of recap matched shots as temporal point annotations [10]. We identify the set of matching shots in the episode, create a binary label vector, and smooth this vec-

tor using a triangular filter. We will refer to these smoothed labels as ground-truth (GT) for story summarization. Extending the supervision helps the model identify meaningful, contiguous sub-stories rather than focusing solely on specific shots highlighted in the recap. For example, it is unlikely that shot s_i is important to the story while $s_{i\pm1}$ is entirely irrelevant (except at scene boundaries). Thus, smoothing is essential to clarify the distinction between positive (essential) and negative (unimportant) shots. Please refer to the supplement for details on the label creation process.

A similar approach can be adopted for dialog utterances. We are able to match 88% of recap utterances to dialog within the smoothed video labels. The rest do not appear in episode $\mathcal{E}_n$ or are picked from extra recorded footage. For simplicity, we inherit labels for the dialogs based on the smoothed label for the temporally co-occurring shot.

4. Method: TaleSumm

We introduce TaleSumm, a two-level hierarchical model that identifies important sub-stories in a TV episode's narrative (illustrated in Fig. 2). At the first level, our approach exploits frame-level (word-level) interactions to extract shot (dialog) representations (Sec. 4.2, Fig. 2(B, C)). At the second level, we capture cross-modal interactions across the entire episode through a Transformer encoder (Sec. 4.3, Fig. 2(A, D)). Before diving into the architecture, we formalize story summarization and introduce notation.

4.1. Problem Statement

Our aim is to extract a multimodal story summary (video and text) from a given episode, typically lasting around 40 minutes, and encompassing multiple key events.

Notation. An episode $\mathcal{E} = (\mathcal{S}, \mathcal{U})$ consists of a set of N video shots $\mathcal{S} = \{s_i\}_{i=1}^{N}$ and a set of dialog utterances $\mathcal{U} = \{u_l\}_{l=1}^{M}$. A *shot* serves as a basic unit of video processing and comprises temporally contiguous frames taken from the same camera, while a *dialog utterance* typically refers to a sentence uttered by an individual as part of a larger conversation. We denote each shot as $s_i = \{f_{ij}\}_{j=1}^{T_i}$, where f_{ij} are sub-sampled frames, and each utterance as $u_l = \{w_{lp}\}_{p=1}^{T_l}$ with multiple word tokens w_{lp}.

Summarization as importance scoring. While humans may naturally select start and end temporal boundaries to indicate important sub-stories, for ease of computation, we discretize time and associate an importance score with each video shot or dialog utterance. Thus, given an episode $\mathcal{E} = (\mathcal{S}, \mathcal{U})$, we formulate story summarization as a binary classification task applied to each element (shot or dialog). The ground-truth labels can be denoted as $\mathbf{y}^{\mathcal{S}} = \{y_i^{\mathcal{S}}\}_{i=1}^{N}$ and $\mathbf{y}^{\mathcal{U}} = \{y_l^{\mathcal{U}}\}_{l=1}^{M}$, where each $y_i^{\mathcal{S}}, y_l^{\mathcal{U}} \in [0, 1]$, signaling their importance to the story summary.

4.2. Level 1: Shot and Dialog Representations

In narrative video production, *shots* play an important role in advancing the storyline and contextualizing neighboring content. We obtain shot-level representations from granular frame-level features to determine how well the shot can contribute to understanding the storyline.

Feature extraction. To capture various aspects of the shot, we use *three* pretrained backbones that capture visual diversity through people, their actions, objects, places, and scenes: $\phi_{\mathcal{S}}^{k}(\cdot), k = \{1, 2, 3\}$. We extract relevant visual information from frame(s) of a given shot, s_i as follows:

$$\mathbf{f}_{ij}^{k} = \phi_{\mathcal{S}}^{k}\left(\{f_{ij}\}\right), \quad \mathbf{f}_{ij}^{k} \in \mathbb{R}^{D_{\mathcal{S}}^{k}}. \tag{1}$$

Note that the backbone may encode a single frame f_{ij} or a short sequence around f_{ij}.

For dialog utterances, we adopt a fine-tuned language model $\phi_{\mathcal{U}}^{\text{FT}}$, to compute contextual word-level features:

$$\mathbf{w}_{lp} = \phi_{\mathcal{U}}^{\text{FT}}\left(\{w_{lp}\}\right), \quad \mathbf{w}_{lp} \in \mathbb{R}^{D_{\mathcal{U}}}. \tag{2}$$

Shot CLS **pooling.** To compute an aggregated shot representation, we combine frame-level signals into a compact representation. An attention-based aggregation (⊞) (inspired by [27]), effectively weighs the most pertinent information (*e.g.* action in a motion-heavy shot or scenery in an establishing shot). First, the frame features from different backbones are projected to the same space (using $\mathbf{W}_{\mathcal{S}}^{k} \in \mathbb{R}^{D \times D_{\mathcal{S}}^{k}}$) and then concatenated to form $\hat{\mathbf{f}}_{ij}^{1:3} \in \mathbb{R}^{3D}$ (Eq. 3). A linear layer $\mathbf{W}_{\mathcal{P}} \in \mathbb{R}^{3\times 3D}$ followed by $\tanh$ and $\mathsf{softmax}$ computes scalar importance scores that are used for weighted fusion:

$$\hat{\mathbf{f}}_{ij}^{1:3} = [W_{\mathcal{S}}^{1}\mathbf{f}_{ij}^{1}, W_{\mathcal{S}}^{2}\mathbf{f}_{ij}^{2}, W_{\mathcal{S}}^{3}\mathbf{f}_{ij}^{3}], \tag{3}$$

$$\boldsymbol{\alpha}_{ij}^{1:3} = \mathsf{softmax}(\tanh(W_{\mathcal{P}}\hat{\mathbf{f}}_{ij}^{1:3})), \tag{4}$$

$$\mathbf{F}_{ij} = \alpha_{ij}^{1}\hat{\mathbf{f}}_{ij}^{1} + \alpha_{ij}^{2}\hat{\mathbf{f}}_{ij}^{2} + \alpha_{ij}^{3}\hat{\mathbf{f}}_{ij}^{3}. \tag{5}$$

We omit bias for brevity. We add relative frame position to $\mathbf{F}_{ij}$ through a *time-embedding* vector, $\mathbf{E}_{j}^{\mathcal{S}}$, similar to Fourier position encoding [84].

A shot transformer [84] ST is used to encode the frame-level feature sequence $\{\mathbf{F}_{ij}\}_{j=1}^{T_i}$. We tap the output from the CLS token appended at the beginning of the sequence (*e.g.* similar to BERT [12]) as the final shot representation:

$$\mathbf{s}_i = \mathsf{ST}(\{\mathbf{F}_{ij} + \mathbf{E}_{j}^{\mathcal{S}}\}_{j=1}^{T_i}), \quad \mathbf{s}_i \in \mathbb{R}^{D}. \tag{6}$$

Dialog utterance representation. First, we project the tokens $\mathbf{w}_{lp}$ to $\mathbb{R}^{D}$ using a linear layer $\mathbf{W}_{\mathcal{U}} \in \mathbb{R}^{D \times D_{\mathcal{U}}}$. As the tokens are already contextualized by $\phi_{\mathcal{U}}^{\text{FT}}$, a simple mean-pool across the p tokens is found to work well:

$$\mathbf{u}_l = \text{mean}_p(\{\mathbf{W}_{\mathcal{U}}\mathbf{w}_{lp}\}_{p=1}^{T_l}), \quad \mathbf{u}_l \in \mathbb{R}^{D}. \tag{7}$$

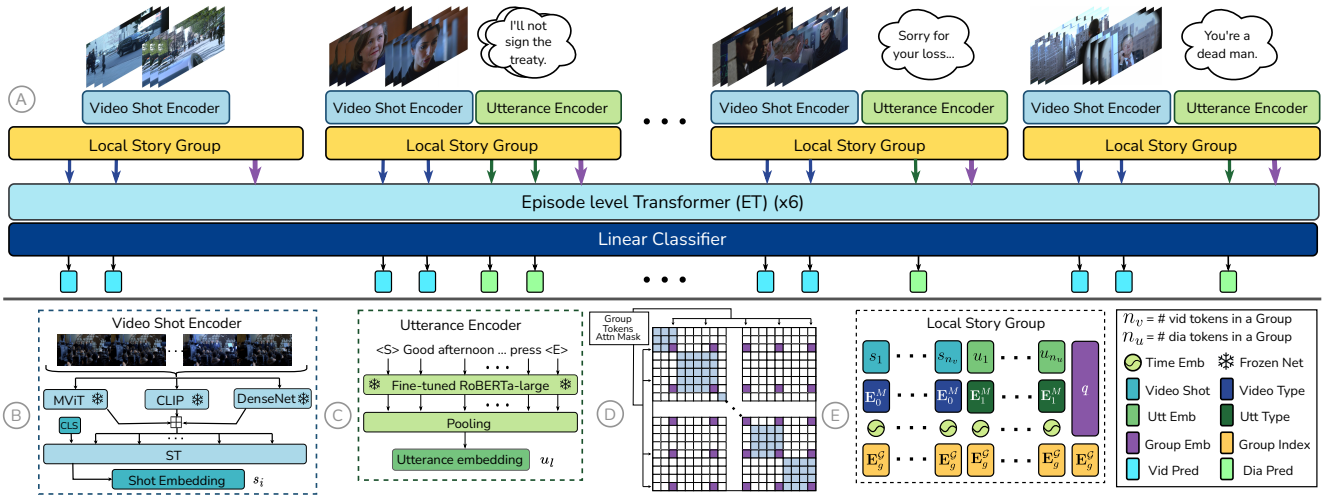

Figure 2. **(A) TaleSumm** ingests all video shots and dialogs of the episode and encodes them using (B) and (C). Based on temporal order, we combine tokens into local story groups (*illustration* shows small groups of 2 shots and 0-2 utterances). To each group, we append a group token and add multiple embeddings, before feeding them to the the episode-level Transformer (ET). For each shot or dialog token, a linear classifier predicts its importance. **(B) Video shot encoder.** For each frame, representations from multiple backbones are fused using attention (⊞). We feed these to a shot Transformer encoder ST, and tap a shot-level representation from the CLS token. **(C) Utterance encoder** uses a fine-tuned language model and avg-pooling across all words of the utterance. **(D) Self-attention mask** illustrates the block-diagonal self-attention structure across the episode. Group tokens across the episode (purple squares) communicate with each other. **(E) Multiple embeddings** are added to the tokens to capture modality type, time, and membership to a local story group.

4.3. Level 2: Episode-level Interactions

We propose an episode-level Transformer encoder, ET, that models interactions across shots and dialog of the entire episode. Predicting the importance of an element (shot or dialog) requires context in a neighborhood; *e.g.* shot in a scene, dialog utterance in a conversation.

Additional embeddings. We arrange shot and dialog tokens based on their order in the episode (see Fig. 2(A)). Learnable *type embeddings* help the model distinguish between shot and dialog modalities ($\mathbf{E}^M \in \mathbb{R}^{2\times D}$). We encode the real time (in seconds) of appearance of each element (shot or dialog) using a binning strategy. Given an episode of T seconds, we initialize Fourier position encodings $\mathbf{E}^T \in \mathbb{R}^{\lceil T/\tau \rceil \times D}$ where τ is the bin-size. Based on the mid-timestamp of each element t, we add $\mathbf{E}^T_t$ to the representation, the $\lfloor t/\tau \rfloor^{\text{th}}$ row in the position encoding matrix. Such time embeddings allow our model to: (i) implicitly encode shot duration; and (ii) relate co-occurring dialogs with video shots without the need for complex attention maps.

Local story groups. The total number of video shots and dialog that make up the sequence length for ET is $S{=}N{+}M$ ($\sim$1500). Self-attention over so many tokens is not only computationally demanding, but also difficult to train due to unrelated temporally distant tokens that happen to look similar. We adopt a block-diagonal attention mask to constrain the tokens to attend to local story regions:

$$\mathbf{A}_{S\times S} = \text{diag}(\mathbb{1}_{n_1\times n_1}, \ldots, \mathbb{1}_{n_g\times n_g}, \ldots, \mathbb{1}_{n_G\times n_G}), \quad (8)$$

where $\mathbb{1}_{n_g\times n_g}$ denotes an all one matrix, n_g is the # of tokens in the g^{th} local block, $\sum_{g=1}^{G} n_g = S$, and diag(...) constructs a block diagonal matrix. We add new learnable group index embeddings $\mathbf{E}^{\mathcal{G}} \in \mathbb{R}^{G\times D}$ to our tokens to inform our model about their group membership.

Group tokens. While capturing interactions across all tokens may lead to poor performance, self-attention only within the local story groups prohibits the model from capturing long-range story dependencies. To enable story group interactions, we propose to add a set of *group tokens* to the input, extending the sequence length to $\hat{S}{=}S{+}G$. The group tokens $\mathbf{q}_g$ represent an additional layer of hierarchy within the episode model as they summarize the story content inside a group and also communicate across groups, providing a way to understand the continuity of the story. Fig. 2(E) shows how group tokens are inserted at the end of each local story group's shot and dialog tokens.

To facilitate cross-group communication, we make two modifications to the self-attention mask: (i) The size of each local group n_g is extended by 1 to incorporate the group token $\mathbf{q}_g$ within the block matrix. We also update $\mathbf{A}$ to reflect this and is of size $\hat{S} \times \hat{S}$. (ii) We compute a binary index $\mathbf{o} \in \{0,1\}^{\hat{S}}$ to represent the locations at which a group token appears in the sequence. The new self-attention mask $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{o}\mathbf{o}^T$ allows group-tokens to communicate. Fig. 2(D) illustrates the attention mask; light blue squares correspond to attention within a group, and sparse purple squares visualize attention across the group tokens.

Importance prediction. We present how shot or dialog scores can be estimated. First, the input tokens to ET are:

$$\hat{\mathbf{s}}_i = \mathbf{s}_i + \mathbf{E}_0^M + \mathbf{E}_{t_i}^T + \mathbf{E}_{g_i}^{\mathcal{G}}, \quad (9)$$

$$\hat{\mathbf{u}}_l = \mathbf{u}_l + \mathbf{E}_1^M + \mathbf{E}_{t_l}^T + \mathbf{E}_{g_l}^{\mathcal{G}}, \quad (10)$$

$$\mathbf{q}_g = \mathbf{q} + \mathbf{E}_g^{\mathcal{G}}. \quad (11)$$

where t_i, t_l and g_i, g_l correspond to the mid-timestamp and group membership of shot s_i and dialog u_l respectively. $\mathbf{q}$ denotes the learnable shared group type embedding.

We feed the updated shot, dialog, and group token representations to ET post LayerNorm [5], a H_{E} layer Transformer encoder with a curated self-attention mask $\hat{\mathbf{A}}$:

$$[\ldots, \tilde{\mathbf{s}}_i, \tilde{\mathbf{u}}_l, \tilde{\mathbf{q}}_g, \ldots] = \mathsf{ET}([\ldots, \hat{\mathbf{s}}_i, \hat{\mathbf{u}}_l, \mathbf{q}_g, \ldots]; \hat{\mathbf{A}}), \quad (12)$$

with all tokens, *i.e.* $\{i\}_1^N$, $\{l\}_1^M$, and $\{g\}_1^G$.

After ET, we compute shot and dialog importance scores using a shared linear classifier $\mathbf{W}^C \in \mathbb{R}^{1\times D}$ followed by sigmoid function $\sigma(\cdot)$:

$$\hat{y}_i^{\mathcal{S}} = \sigma(\mathbf{W}^C \tilde{\mathbf{s}}_i) \text{ and } \hat{y}_l^{\mathcal{U}} = \sigma(\mathbf{W}^C \tilde{\mathbf{u}}_l)\, \forall i, l\,. \quad (13)$$

4.4. Training and Inference

Training. TaleSumm is trained in an end-to-end fashion with *BinaryCrossEntropy* (BCE) loss. We provide positive weights, w (ratio of negatives to positives) to account for class imbalance. Modality specific losses are added:

$$\mathcal{L} = \mathsf{BCE}\left(\hat{\mathbf{y}}^{\mathcal{S}}, \mathbf{y}^{\mathcal{S}}; w^{\mathcal{S}}\right) + \mathsf{BCE}\left(\hat{\mathbf{y}}^{\mathcal{U}}, \mathbf{y}^{\mathcal{U}}; w^{\mathcal{U}}\right). \quad (14)$$

Inference. At test time, we follow the procedure outlined in Sec. 4.3 and generate importance scores for each video shot and dialog utterance (Eq. 13).

Model ablations. As we will see empirically, our model is versatile and well-suited for adding/removing modalities or additional representations by adjusting the sequence length of the Transformer (number of tokens). It can also be modified to act as an unimodal model that applies only to video or dialog utterances by disregarding other modalities.

5. Experiments and Analysis

We first discuss the experimental setup.

Data splits. We adopt 3 settings. (i) IntraCVT: On *24*, most experiments (Tabs. 3 to 5) follow an intra-season 5-fold cross-validation-test strategy. (ii) X-Season: On *24*, we assess cross-season generalization using a 7-fold cross-validation-test (Tab. 7). (iii) X-Series: shows transfer results from *24* to *PB* (Tab. 7). Details in the supplement.

Evaluation metric. We adopt Average Precision (AP, area under PR curve) as the metric to compare predicted importance scores of shots or dialogs against ground-truth.

	Video-only					Dialog-only		
	Avg	Max	Cat	Tok	⊞	Max	Avg	wCLS
MLP	42.3	42.3	42.3	42.4	42.5	35.7	35.7	35.8
woG + FA	51.8	51.9	51.1	51.6	52.0	44.5	44.5	44.6
wG + SA	52.4	52.5	53.3	53.3	**53.4**	46.5	*46.5*	**47.2**

Table 3. **Rows** demonstrate methods for capturing episode-level interactions. In an MLP, tokens are independent. woG+FA is a Transformer encoder that captures full-attention over the entire episode without grouping; and wG+SA uses the proposed architecture with local story groups and sparse-attention. **Columns** describe the *aggregation* method used to combine frame (or token) level features into shot (or utterance) representation. C1 and C7 use average pooling. C2 and C6 use max pooling. C3-C5 are variants of ST: C3 concatenates backbone features of each frame, C4 uses backbone features as separate tokens, and C5 uses proposed ⊞ attention fusion. C8 uses the CLS token for dialog. Chosen: ⊞ for shot, and average pooling for utterance representation.

5.1. Implementation details

We present some high-level details here.

Feature backbones. We adopt three visual backbones: DenseNet169 [29] for object understanding; MViT [13] for action information; and OpenAI CLIP [65] for semantics.

To encode dialog, we adapt RoBERTa-large [99] for extractive summarization using parameter-efficient fine-tuning [28, 55] on the text from our dataset. The backbone is frozen when training TaleSumm for story summarization.

Additional backbone details are in the supplement.

Architecture. We find H_{S}=1, H_{E}=6, and n_g=20 to work best. Both ST and ET have the same configuration: 8 attention heads and D=128. We tried several architecture configurations, details are in the supplement.

Training details. We *randomly* sample up to 25 frames per shot during training as a form of data augmentation and use *uniform* sampling during inference. Our model is implemented in PyTorch [61], has 1.94 M parameters, and is trained on 4 RTX-2080 Ti GPUs for a batch size of 4 (*i.e.* 4 entire episodes). The optimizer, learning rate, dropout, and other hyperparameters are tuned for best performance on the validation set and indicated in the supplement.

5.2. Experiments on *24*

Architecture ablations. Results in Tab. 3 are across two dimensions: (i) columns span the shot or utterance level and (ii) rows span the episode level. All model variants outperform a baseline that predicts a random score between $[0, 1]$: AP 34.2 (video) and 30.4 (dialog), over 1000 trials.

Across rows, we observe that the MLP performs worse than the other two variants by almost 10% AP score because assuming independence between story elements is bad. Our proposed approach with local story groups and

	Model	AttnMask	GToken	Video AP	Dialog AP
1	Video-only	SA	✓	53.4 ± 3.9	-
2	Dialog-only	SA	✓	-	47.2 ± 3.9
3	TaleSumm	FA	-	51.8 ± 3.6	43.8 ± 4.7
4		FA	✓	51.9 ± 3.7	44.0 ± 4.6
5		SA	-	53.9 ± 3.4	48.8 ± 4.6
6		SA	✓	**54.2** ± 3.3	**49.0** ± 4.9

Table 4. TaleSumm ablations. The *AttnMask* column indicates if the self-attention mask is applied over the full episode (FA) or uses sparse block diagonal structure of story groups (SA). The *GToken* indicates whether the group token is absent (-) or present (✓). R6 is our final chosen model for subsequent experiments.

Model	Val		Test	
	Video AP	Dialog AP	Video AP	Dialog AP
PGLSUM [3]	48.8 ± 3.3	-	47.1 ± 2.4	-
MSVA [17]	47.3 ± 3.8	-	45.5 ± 1.2	-
Video-Only	53.4 ± 3.9	-	**50.6** ± 3.6	-
PreSumm [45]	-	43.1 ± 3.3	-	41.6 ± 2.0
Dialog-Only	-	47.2 ± 3.9	-	43.4 ± 2.8
A2Summ [25]	35.1 ± 1.8	33.2 ± 2.8	33.8 ± 1.7	31.6 ± 2.2
TaleSumm (Ours)	**54.2** ± 3.3	**49.0** ± 4.9	*50.1* ± 2.8	**46.0** ± 2.1

Table 5. Comparison against SoTA video-only, text-only, and multimodal summarization models. Our approach outperforms previous work by a significant margin.

sparse-attention (wG+SA) outperforms a vanilla encoder without groups and full-attention (woG+FA) by 1-2% on the video model and 2-3% for the dialog model.

Across columns, performance changes are minor. However, when using wG+SA at the episode-level, gated attention fusion with a shot transformer (⊞) improves results over Avg and Max pooling by 1%. For dialog-only, though wCLS outperforms Avg and Max by 0.7%, we adopt Avg pooling for its effective performance in a multimodal setup.

TaleSumm ablations are presented in Tab. 4. Rows 1 and 2 highlight the best video-only and dialog-only models (from Tab. 3). We report mean ± std dev on the val set. Std dev is found to be high due to variation across multiple folds; but low across random seeds. Results for joint prediction of video shot and utterance importance are shown in rows 3-6. Our proposed approach in row 6 performs best for both modalities, outperforming rows 3-5.

SoTA comparison. We compare against SoTA methods: (i) video-only (PGLSUM [3], MSVA [17]), (ii) dialog-only (PreSumm [45]), and (iii) multimodal (A2Summ [25]) in Tab. 5. While none of the above methods are built for processing 40 minutes of video, we make modifications to them to make them comparable to our work (details in the supplement). On both the validation and the test set, TaleSumm outperforms all other baselines in both modalities.

Model	SumMe			TVSum		
	F1	SP	KT	F1	SP	KT
MSVA [17]	52.4	12.3	9.2	*63.9*	32.1	22.0
PGLSUM [3]	56.2	17.3	12.7	*63.9*	**40.5**	**28.2**
A2Summ [25]	54.0	3.5	2.8	62.9	25.2	17.1
TaleSumm (Ours)	**57.5**	**23.8**	**17.6**	**64.0**	26.7	18.2

Table 6. Comparison with SoTA methods on the SumMe [21] and TVSum [75] benchmark datasets. Metrics are suggested by Otani *et al*. [54]: F1, Kendall's τ (KT), and Spearman's ρ (SP).

	Model	X-Season (*24*)		X-Series (*PB*)	
		Video	Dialog	Video	Dialog
1	MSVA [17]	46.7 ± 2.7	-	32.7	-
2	PGLSUM [3]	47.1 ± 2.4	-	34.5	-
3	PreSumm [45]	-	41.3 ± 3.2	-	**38.3**
4	A2Summ [25]	33.5 ± 3.2	31.7 ± 2.9	20.2	19.0
5	TaleSumm (Ours)	**51.0** ± 4.6	**46.0** ± 5.5	**36.7**	*35.7*

Table 7. We evaluate our model's generalization across seasons within *24* and across TV shows (*24* → *Prison Break*). R5 showcases superiority of our methods compared to SoTA. In X-Series, the random baseline achieves 21.3 and 19.1 for video and dialog.

5.3. Analysis and Discussion

Video summarization benchmarks. We evaluate TaleSumm on SumMe [21] and TVSum [75]. However, both datasets are small (25 and 50 videos) and have short duration videos (few minutes). As splits and metrics are not comparable across previous works, we re-ran the baselines.

While MSVA uses three feature sets: i3d-rgb, i3d-flow [7] and GoogleNet [78] with intermediate fusion, PGLSUM uses GoogleNet and captures local and global features. In contrast, A2Summ [25] aligns cross-modal information using dual-contrastive loss between video (GoogleNet features) and text (captions generated using GPT-2 [64], embedded by RoBERTa [99] at frame level).

Similar to MSVA, we fuse all 3 features. Even though TaleSumm is built for long videos (group blocks, sparse attention), Tab. 6 shows that we achieve SoTA on SumMe. The drop in performance on TVSum may be due to video diversity (documentaries, how-to videos, *etc*.).

Generalization to a new season/TV series. Tab. 7 shows results in two different setups. In X-Season, we see the impact of evaluating on unseen seasons (in a 7-fold cross-val-test). While TaleSumm outperforms baselines, it is interesting that most methods show comparable performance across IntraCVT and X-Season setups (see Tabs. 5 and 7).

In the X-Series setting, we train our model on *24* and evaluate on *Prison Break*. Although both series are crime thrillers, there are significant visual and editing differences between the two shows. Our approach obtains good scores on video summarization, and is a close second on dialog.

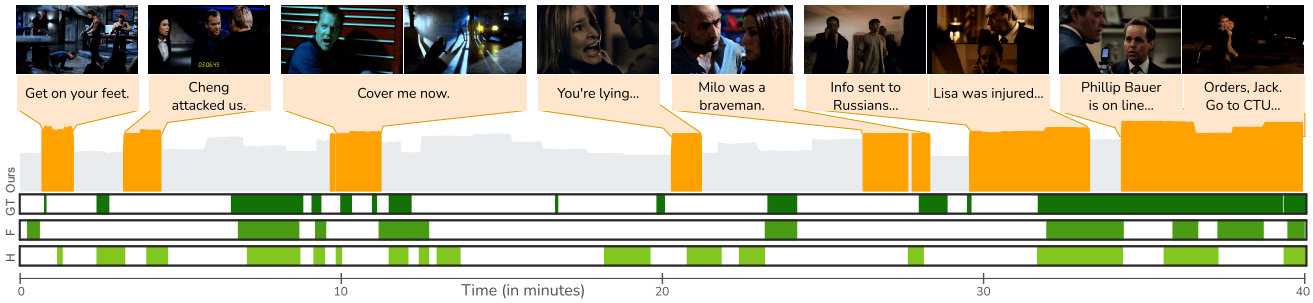

Figure 3. TaleSumm predictions on S06E22 of *24* (test set). "Ours" filled-plot illustrates the importance score profile over time, where orange patches indicate story segments selected for summarization. Annotations are shown below: ground-truth (GT), fandom (F), and human annotated (H). The story: Amid the high-stakes sequence depicted in the selected groups 1-3, Zhou Yong's team captures Josh Bauer, leading to a firefight with Jack Bauer, who seeks Josh's location. Negotiations with Phillip Bauer over Josh's return for a vital circuit board escalate global tensions between Russia and the USA. Simultaneously, Mike Doyle defies Jack's wishes and departs with Josh by helicopter (segment 7). Parallely, Lisa, backed by Tom Lennox, confronts a Russian agent, leading to her injury (4, 6). Morris attempts to console Nadia for Milo's loss at CTU in 5. Escalating global tensions and the imminent showdown mark the episode.

Dataset	Cronbach α		Pairwise F_1		Fleiss' κ	
	Video	Dialog	Video	Dialog	Video	Dialog
SumMe	0.88	-	0.31	-	0.21	-
TVSum	0.98	-	0.36	-	0.15	-
PlotSnap (Ours)	0.91	0.93	0.59	0.60	0.38	0.39

Table 8. Label consistency across datasets.

Methods	Fandom (F)		Human (H)	
	Video AP	Dialog AP	Video AP	Dialog AP
GT	64.1	63.8	44.8	42.2
PGLSUM	43.0	-	47.6	-
MSVA	34.3	-	42.2	-
PreSumm	-	43.6	-	46.7
A2Summ	28.7	29.7	41.0	41.1
TaleSumm	**44.5**	**45.5**	**48.7**	**50.9**

Table 9. Results on labels from *24* fan site (F) and human-annotated story summaries (H) averaged over 17 episodes of *24*.

Label consistency. As suggested by [21, 75], label consistency is crucial to evaluate summarization methods. We assess PlotSnap using Cronbach's α, pairwise F_1-measure, and another agreement score: Fleiss' κ.

We obtain three sets of labels for 17 episodes of *24* (details in supplement). (i) GT: obtained from matching recaps; (ii) F: maps plot events from a *24* fan site[1] to videos; (iii) H: human response for a summary. Our labels have superior consistency compared to SumMe [21] and TVSum [75] (see Tab. 8), indicating that identifying key story events in a TV episode is less subjective than scoring importance for generic Youtube videos. Tab. 9 shows the results for the baselines and TaleSumm on the two other labels F and H. Our model predictions are better aligned with both labels.

Qualitative analysis. We show the model's predictions and compare against all three labels (GT, F, and H) for one episode in Fig. 3. Our model identifies many important story segments that are also part of the annotations.

6. Conclusion

Our work pioneered the use of TV episode recaps for story understanding. We proposed PlotSnap, a dataset of two TV shows with high-quality recaps, leveraging them for story summarization labels, while showing high consistency across labeling approaches. We introduced TaleSumm, a hierarchical summarization approach that captures and compresses shots and dialog, and enables cross-modal interactions across the entire episode, trainable on a single GPU of 12 GB. We performed thorough ablations, established SoTA performance, and demonstrated transfer across seasons, other series, and even movie genres. For reproducibility and encouraging future work, we will release the code and share the dataset, as keyframes, features, and labels.

Limitations and future work. While our current work focuses on recaps obtained from a limited genre and two TV series, we believe the approach should be scalable to additional genres and datasets. Early experiments in evaluating our model on condensed movies (CMD) [6] show limited improvements. Our approach to story summarization does not explicitly model the presence of characters (*e.g.* via person and face tracks and their emotions [76]) which are central to any story and this can be an important direction for future work. Additional discussions are provided in the supplementary material.

Acknowledgments. We thank the Bank of Baroda for partial travel support, and IIIT-H's faculty seed grant and Adobe Research India for funding. Special thanks to Varun Gupta for assisting with experiments and the Katha-AI group members for user studies.

[1] https://24.fandom.com/wiki/Day_6:_4:00am-5:00am talks about the key story events in S06E22 in a *Previously on 24* section (see Fig. 3).

References

[1] 24 (TV series, IMDb). https://www.imdb.com/title/tt0285331/, 2001. 2, 3

[2] Prison Break (IMDb). https://www.imdb.com/title/tt0455275/, 2005. 2, 3

[3] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining Global and Local Attention with Positional Encoding for Video Summarization. In *IEEE International Symposium on Multimedia (ISM)*, 2021. 7

[4] George Awad, Keith Curtis, Asad A. Butt, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, , and Georges Quénot. An overview on the evaluated video retrieval tasks at TRECVID 2022. In *Proceedings of TRECVID*, 2022. 2, 3

[5] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv: 1607.06450*, 2016. 6

[6] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed Movies: Story Based Retrieval with Contextual Embeddings, 2020. 8

[7] João Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[8] Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. SummScreen: A Dataset for Abstractive Screenplay Summarization. In *Association of Computational Linguistics (ACL)*, 2022. 2, 3

[9] Shixing Chen, Chun-Hao Liu, Xiang Hao, Xiaohan Nie, Maxim Arap, and Raffay Hamid. Movies2Scenes: Using movie metadata to learn scene representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[10] Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A Flexible Model for Training Action Localization with Varying Levels of Supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[11] Sandra Eliza Fontes de Avila, Ana Paula Brandão Lopes, Antonio da Luz, and Arnaldo de Albuquerque Araújo. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011. 2

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805*, 2018. 4

[13] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *International Conference on Computer Vision (ICCV)*, 2021. 6

[14] Cheng-Yang Fu, Joon Lee, Mohit Bansal, and Alexander Berg. Video Highlight Prediction Using Audience Chat Reactions. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2017. 3

[15] Tsu-Jui Fu, Shao-Heng Tai, and Hwann-Tzong Chen. Attentive and adversarial learning for video summarization. In *Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2

[16] Xiyan Fu, Jun Wang, and Zhenglu Yang. MM-AVS: A Full-Scale Dataset for Multi-modal Summarization. In *North American Chapter of Association of Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021. 2

[17] Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. Supervised Video Summarization Via Multiple Feature Sets with Parallel Attention. *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6s, 2021. 7

[18] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019. 3

[19] Dan B Goldman, Brian Curless, David Salesin, and Steven M. Seitz. Schematic Storyboarding for Video Visualization and Editing. page 862–871, 2006. 2

[20] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision Workshops (ECCVW)*. Springer, 2014. 2

[21] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision Workshops (ECCVW)*, 2014. 2, 3, 7, 8

[22] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[23] Xiang Hao, Kripa Chettiar, Ben Cheung, Vernon Germano, and Raffay Hamid. Intro and Recap Detection for Movies and TV Series. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3

[24] Monica-Laura Haurilet, Makarand Tapaswi, Ziad Al-Halah, and Rainer Stiefelhagen. Naming TV characters by watching and analyzing dialogs. In *Winter Conference on Applications of Computer Vision (WACV)*, 2016. 3

[25] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and Attend: Multimodal Summarization with Dual Contrastive Losses. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 7

[26] Hsuan-I Ho, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Summarizing First-Person Videos from Third Persons' Points of Views. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018. 3

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, (8):1735–1780, 1997. 4

[28] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*, 2019. 6

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[30] Jia-Hong Huang and Marcel Worring. Query-controllable Video Summarization. In *International Conference on Multimedia Retrieval (ICMR)*, 2020. 3

[31] M. Islam, M. Hasan, K. Athrey, T. Braskich, and G. Bertasius. Efficient Movie Scene Detection using State-Space Transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[32] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *European Conference on Computer Vision Workshops (ECCVW)*, 2022. 3

[33] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. Neural Extractive Summarization with Hierarchical Attentive Heterogeneous Graph Network. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020. 2

[34] Vivekraj V. K., Debashis Sen, and Balasubramanian Raman. Video Skimming: Taxonomy and Comprehensive Survey. *ACM Comput. Surv.*, 2019. 2

[35] Zeeshan Khan, C.V. Jawahar, and Makarand Tapaswi. Grounded Video Situation Recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[36] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-Scale Video Summarization Using Web-Image Priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[37] Gunhee Kim, Leonid Sigal, and Eric P. Xing. Joint Summarization of Large-Scale Collections of Web Images and Videos for Storyline Reconstruction. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[38] Johannes Kopf, Michael F. Cohen, and Richard Szeliski. First-Person Hyper-Lapse Videos. *ACM Transactions on Graphics*, 2014. 2

[39] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012. 2

[40] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[41] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. TVQA: Localized, Compositional Video Question Answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 3

[42] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. TVQA+: Spatio-Temporal Grounding for Video Question Answering. In *Association of Computational Linguistics (ACL)*, 2020. 3

[43] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Association of Computational Linguistics (ACL)*, 2020. 2

[44] Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. VideoXum: Cross-modal Visual and Textural Summarization of Videos. *IEEE Transactions on Multimedia*, 2023. 2, 3

[45] Yang Liu and Mirella Lapata. Text Summarization with Pretrained Encoders. In *Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. 2, 7

[46] Zheng Lu and Kristen Grauman. Story-Driven Summarization for Egocentric Video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[47] Mingyang Ma, Shaohui Mei, Shuai Wan, Zhiyong Wang, David Dagan Feng, and Mohammed Bennamoun. Similarity Based Block Sparse Subset Selection for Video Summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3967–3980, 2021. 2

[48] Arsha Nagrani and Andrew Zisserman. From Benedict Cumberbatch to Sherlock Holmes: Character Identification in TV series without a Script. In *British Machine Vision Conference (BMVC)*, 2017. 3

[49] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Learnable PINs: Cross-Modal Embeddings for Person Identity. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018. 3

[50] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar Gulçehre, and Bing Xian. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Computational Natural Language Learning (CoNLL)*, 2016. 2

[51] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, and Bing Xiang. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Computational Natural Language Learning (CoNLL)*, 2016. 2, 3

[52] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-It! Language-guided Video Summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[53] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 2, 3

[54] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkilä. Rethinking the Evaluation of Video Summaries. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[55] Pinelopi Papalampidi and Mirella Lapata. Hierarchical3D Adapters for Long Video-to-text Summarization. *arXiv:2210.04829*, 2022. 2, 3, 6

[56] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie Plot Analysis via Turning Point Identification. In *Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. 2

[57] Pinelopi Papalampidi, Frank Keller, Lea Frermann, and Mirella Lapata. Screenplay Summarization Using Latent Narrative Structure. In *Association of Computational Linguistics (ACL)*, 2020. 2

[58] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Film trailer generation via task decomposition. *arXiv preprint arXiv:2111.08774*, 2021. 3

[59] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie Summarization via Sparse Graph Construction. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021. 3

[60] Jae Sung Park, Trevor Darrell, and Anna Rohrbach. Identity-Aware Multi-Sentence Video Description. In *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 3

[61] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 6

[62] Stefano Pini, Marcella Cornia, Federico Bolelli, Lorenzo Baraldi, and Rita Cucchiara. M-VAD Names: A Dataset for Video Captioning with Naming. *Multimedia Tools Appl.*, page 14007–14027, 2019. 3

[63] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. Nonchronological Video Synopsis and Indexing. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2008. 2

[64] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 7

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*. PMLR, 2021. 6

[66] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020. 2

[67] Mrigank Rochan and Yang Wang. Video summarization by learning from unpaired data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[68] Anna Rohrbach, Marcus Rohrbach, Weijian Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent Multi-sentence Video Description with Variable Level of Detail. In *German Conference on Pattern Recognition (GCPR)*, 2014. 3

[69] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for Movie Description. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[70] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie Description. *International Journal of Computer Vision (IJCV)*, 123(1):94–120, 2017. 3

[71] Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. Visual Semantic Role Labeling for Video Understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[72] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: A Large-scale Dataset for Multimodal Language Understanding. In *Advances in Neural Information Processing Systems-Workshop (NeurIPS-W)*, 2018. 2, 3

[73] K. Pramod Sankar, C. V. Jawahar, and Andrew Zisserman. Subtitle-free Movie to Script Alignment. In *British Machine Vision Conference (BMVC)*, 2009. 3

[74] Yassir Saquil, Da Chen, Yuan He, Chuan Li, and Yong-Liang Yang. Multiple Pairwise Ranking Networks for Personalized Video Summarization. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[75] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TVSum: Summarizing web videos using titles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3, 7, 8

[76] Dhruv Srivastava, Aditya Kumar Singh, and Makarand Tapaswi. How you feelin'? Learning Emotions and Mental States in Movie Scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

[77] Min Sun, Ali Farhadi, Ben Taskar, and Steve Seitz. Salient montages from unconstrained videos. In *European Conference on Computer Vision Workshops (ECCVW)*, 2014. 2

[78] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 7

[79] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Story-Based Video Retrieval in TV Series Using Plot Synopses. In *International Conference on Multimedia Retrieval (ICMR)*, 2014. 3

[80] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. StoryGraphs: Visualizing Character Interactions as a Timeline. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[81] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Aligning plot synopses to videos for story-based retrieval. *International Journal of Multimedia Information Retrieval (IJMIR)*, 4(1):3–16, 2015. 3

[82] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[83] Tengda Han and Max Bain and Arsha Nagrani and Gül Varol and Weidi Xie and Andrew Zisserman. AutoAD II: The Sequel - Who, When, and What in Movie Audio Description. In *International Conference on Computer Vision (ICCV)*, 2023. 3

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 4

[85] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. MovieGraphs: Towards Understanding Human-Centric Situations from Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[86] W. Wolf. Key Frame Selection by Motion Analysis. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1996. 2

[87] F. Xiao, K. Kundu, J. Tighe, and D. Modolo. Hierarchical Self-supervised Representation Learning for Movie Understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[88] Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun. Title Generation for User Generated Videos. In *European Conference on Computer Vision Workshops (ECCVW)*, 2016. 3

[89] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning (ICML)*. PMLR, 2020. 2

[90] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Summary Transfer: Exemplar-based Subset Selection for Video Summarization. *CoRR*, abs/1603.03369, 2016. 2

[91] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European Conference on Computer Vision Workshops (ECCVW)*, 2016. 2

[92] Ke Zhang, Kristen Grauman, and Fei Sha. Retrospective encoders for video summarization. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018. 2

[93] Yujia Zhang, Michael Kampffmeyer, Xiaoguang Zhao, and Min Tan. DTR-GAN: Dilated temporal relational adversarial network for video summarization. In *Proceedings of the ACM Turing Celebration Conference-China*, 2019. 2

[94] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive Summarization as Text Matching. In *Association of Computational Linguistics (ACL)*, 2020. 2

[95] Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. MSMO: Multimodal Summarization with Multimodal Output. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 2

[96] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. DSNet: A Flexible Detect-to-Summarize Network for Video Summarization. *IEEE Transactions on Image Processing*, 30: 948–962, 2021. 2

[97] Wencheng Zhu, Yucheng Han, Jiwen Lu, and Jie Zhou. Relational reasoning over spatial-temporal graphs for video summarization. *IEEE Transactions on Image Processing*, 31: 3017–3031, 2022. 2

[98] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[99] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A Robustly Optimized BERT Pre-training Approach with Post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, 2021. 6, 7