NarrAD: Automatic Generation of Audio Descriptions for Movies with Rich Narrative Context

Jaehyeong Park Juncheol Ye Seungkook Lee Hyun W. Ka Dongsu Han
KAIST, Republic of Korea
{woguddlrj676,juncheol,sklee07074,hyun.ka,dhan.ee}@kaist.ac.kr

Abstract

Audio Description (AD) is a narration designed to enhance accessibility for visually impaired individuals by conveying the key visual elements of a video. Thus, automating AD generation for long-form videos, such as movies and dramas, provides high social value but is a challenging task. First, AD must reflect the narrative context of the entire movie, including the storyline, names of characters and places, and the cultural setting. Second, to avoid disrupting the immersive experience of the movie, AD must not overlap with the characters' dialogues, requiring the delivery of numerous visual elements in concise sentences. This paper presents NarrAD, a training-free AD generation framework that satisfies both of the requirements by leveraging rich narrative context in movie scripts and curating information across narration slots. Experiments on the MAD dataset demonstrate that our approach outperforms prior works in both captioning and LLM-based metrics. In the user study with 600 subjects, NarrAD achieves the highest user experience and movie comprehension. NarrAD's AD samples are available at https://bit.ly/4aSwOTr.

1. Introduction

Audio Description (AD), designed to bridge the sensory gap for visually impaired viewers, deliver important visual features beyond dialogue, such as characters' actions, attire, scene settings, and contextual nuances. Studies have shown that AD enriches video comprehension for many sighted audiences [41], highlighting its social value. However, the availability of AD for long-form video is significantly low—social organizations [1] struggle to provide AD for 11,516 titles, a stark contrast to the 811,306 total movie titles listed on IMDb. This availability gap is largely due to the prohibitive production cost ($1,800 per hour), which relies on professional writers [24].

Automatic generation of AD offers a cost-effective alternative. However, generating AD for long-form videos, such as movies and dramas, is much more challenging than typical video captioning due to their distinct requirements [1]:

- The AD must incorporate narrative context to provide a consistent and immersive experience. The narrative context of a movie includes the plot, the names of the characters, locations, and key elements of the movie. However, generating narrative context often requires understanding of the entire movie beyond understanding a short sequence of movements.
- The AD should be inserted only during pauses in audio in a movie because overlapping ADs with a dialogue significantly undermines the auditory experience. A major challenge is that the narration slots allocated for AD are often too short to fully describe all the visual elements of a given moment. Thus, it is crucial to identify and prioritize text that has important semantics to minimize information loss during summarization.

Previous works [14–16,52,57] attempt to capture the narrative context using a variety of sources, including subtitles, ADs generated for previous scenes, and/or an external character bank. However, these short-term temporal contexts are insufficient to capture the narrative context that is scattered throughout an entire movie. Moreover, they generate ADs of uniform length, significantly impairing the immersive user experience because the narration interferes with the dialogue [57] or contains too little information [52].

To address this, we present NarrAD, an automated AD generation framework that conveys as much rich contextual information as possible within the given narration slot (i.e., pauses in audio). Our key insight is that movie scripts provide a detailed and concentrated narrative context, including the director's intent on what should be conveyed to the audience. Thus, NarrAD provides movie scripts as context to a multimodal large language model (LLM). By instructing it to describe the video using the script provided, we generate ADs enriched with detailed context. However, there are two challenges that must be addressed:

First, naively using the entire movie script results in poor AD quality. To generate accurate descriptions, we must precisely align movies and their scripts to extract the part related to the given video segment from the script. To enable

2642-9381/25/$31.00 ©2025 IEEE
DOI 10.1109/WACV61041.2025.00050

this, we propose a video-to-script retrieval algorithm that operates at the *scene* level through dialogue synchronization and scene recognition.

Second, ADs generated by the LLM are often too long to fit in the given narration slot because of the numerous visual elements in a video segment. As a result, when trying to shorten the length of AD, information loss can occur. To minimize the loss, we leverage the fact that some information, such as the spatial and temporal background, and the appearance of characters do not change frequently, so they reappear across multiple video segments. Thus, we carefully curate ADs across multiple slots, prioritizing the removal of overlapping information to convey as much information as possible within the given slots.

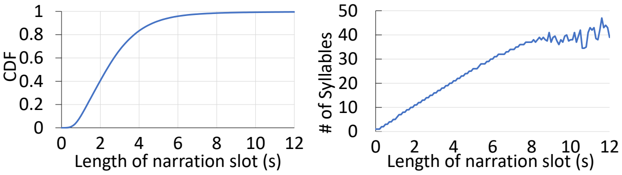
In conclusion, we make the following contributions: (1) NarrAD is the first system to use movie scripts to automatically generate ADs. We demonstrate that the approach is cost-effective and that the rich narrative context provided by the movie scripts enhances the quality of ADs. (2) Using our information curating algorithm, we generate ADs that fit the given narration slots, while maximizing information delivery. (3) NarrAD offers a training-free approach that not only outperforms prior training-free designs, but also fine-tuning-based approaches on the MAD dataset [44], in both standard captioning metrics [4,27,48] and LLM-based metrics [16,57]. (4) Evaluation with 600 human subjects shows that NarrAD enhances user experience and movie comprehension.

2. Background & Related Work

Problem definition. Suppose there is a video V and timestamps T, which indicate the narration slots where AD needs to be inserted. AD generation is a task that creates a description U_i for the video segment v_i, corresponding to each timestamp $[t_i^{start}, t_i^{end}]$.

Syllable count of AD. The syllable counts in each AD should be adjusted to fit the size of the provided narration slot, as it directly indicates the time required to pronounce a specific sentence. ADs with a syllable count longer than the given narration slot can cause audio overlap. Fig. 1 presents the results of a statistical analysis on the correlation between the length of narration slots and the syllable count of the corresponding ADs in the MAD training set [44]. Ideally, as in this dataset, the two should have a linear relationship. Building on this observation, we utilize the statistics from the MAD training set to estimate the time required to pronounce an AD with a specific syllable count and the appropriate syllable count for the given narration slot.

AD generation. A large body of work has been devoted to automatic generation of ADs [14–16, 20, 28, 46, 52, 56, 57]. AutoAD [15], AutoAD2 [14], and AutoAD3 [16] use pretrained models (GPT-2 and Video-Llama2), enhanced with additional trainable parameters and fine-tuned using the AD dataset. autoAD-Zero [52], MM-Narrator [57], and MM-Vid [28] generate ADs using off-the-shelf multimodal LLM, employing prompt engineering techniques. They utilize past ADs and subtitles to reflect the narrative context of a movie [14,15,57]. The past ADs refer to narrative descriptions generated for the preceding scene in the same movie. In addition, they propose a character bank for naming characters in AD [14, 16, 52, 57].



(a) Length of narration slot

(b) Median syllable counts

Figure 1. The length of narration slots in the MAD dataset [44] along with the corresponding syllable counts.

However, existing approaches fall short of fully capturing the narrative context of movies. Relying on past ADs fails to capture rich narrative context scattered throughout the entire movie, as it depends on past short-term temporal context. Additionally, the recursive use of past ADs introduces error propagation, compounding inaccuracies over time. The character bank fails to include proper nouns other than character names, such as names of places, buildings, or specific objects. Subtitles often do not align with the video's visual elements, and the information obtained from character's dialogues is something visually impaired people can already grasp, thus not aiding AD generation. Furthermore, none of these methods adjust AD sentence length for available narration slots, resulting in uniform length ADs that may overlap with each other or with character's dialogue [57] or convey insufficient information [52].

Utilizing movie script. Prior work utilizes scripts for movie summarization [37–39], action recognition [11, 22, 30, 31, 34], identifying character names [10, 12, 35, 43, 45], and scene recognition [6]. Our work is the first to leverage narrative context from movie scripts specifically for generating ADs.

Multimodal LLMs have achieved notable success in vision-language tasks, such as video captioning [7, 8, 21, 53, 55], video question answering [3, 19, 54], and video understanding [28, 57]. Techniques like multimodal in-context learning [47, 49] and chain-of-thought reasoning [50, 59] demonstrate remarkable performance in utilizing multimodal LLM without the need of fine-tuning. These methods work by providing a task-specific introduction, a few examples, and a series of intermediate reasoning steps into the multimodal LLM as an instruction prompt to solve downstream tasks. In this work, we utilize the multimodal LLM to address tasks for generating AD, including scene recognition, video description, and natural language processing.

Context of LLM	Length	Relevance	SPICE
Past ADs, Script	1 scene	✓	**8.3**
Past ADs, Script	30 scenes	✓	7.2
Past ADs, Script	1 scene	X	6.5
Past ADs	0 scene	X	3.2

Table 1. Variation in AD quality depending on the context provided to the multimodal LLM. Relevance indicates whether the movie script contains content related to the video segment it intends to describe. We use *Signs* (2002) from the MAD dataset [44].

3. Key Insight & Challenge

Utilizing movie scripts as context in LLM. One of the most intuitive ways to obtain detailed narrative context for a movie is to read its script. The stage directions in movie scripts provide detailed descriptions of what is happening in the video, including the names of characters and places, and the temporal/spatial backgrounds. They encapsulate narrative details that must be inferred from fragmented information scattered throughout the movie.

Tab. 1 shows the impact of movie scripts on AD quality. When movie scripts are used as the context in a multimodal LLM, there is a significant improvement in AD quality (SPICE 8.3 vs 3.2). However, we observe that the quality of AD significantly varies depending on the length of the movie scripts and their relevance to the video segment. AD quality is highest when the script context includes a scene that exactly matches the video segment (SPICE: 8.3). Incorporating an unrelated scene (SPICE: 6.5) or providing a script that is too long (SPICE: 7.2), even if it contains the matching scene, significantly reduces the quality. Thus, extracting concise and highly relevant narrative contexts from entire movie scripts is crucial for AD generation.

Challenge #1. Extracting the narrative context relevant to a brief video segment from an entire movie script is challenging. A naïve approach is to align at the sentence level, either by using subtitles as in previous studies [11, 12, 22, 30, 31, 42] or through general video-to-script retrieval methods [5, 17, 18, 25, 26, 33, 40, 51, 58]. However, due to inconsistencies between the movie and its script, along with the repetitive nature of movies, these methods become highly challenging. The video segment may not have an exact match in the movie script, and vice versa because of minor changes during filming and editing. Furthermore, in movies, the same characters and backgrounds often reappear, resulting in many similar shots. We need to precisely match the video segment to the movie script, but due to the numerous similar parts, it becomes challenging to identify the single correct match. These factors make sophisticated sentence-level retrieval exceptionally difficult.

Coordinated information delivery. ADs are not simply a collection of captions for independent video segments but form a sequence of descriptions. Users listen to multiple ADs in succession to understand the movie. Curating information inevitably leads to information loss, but from the perspective of a sequence, it's possible to minimize this loss. We observe that information like spatial and temporal backgrounds, characters' appearances, and long-lasting actions, which do not change frequently, can be repeated across several adjacent ADs. When numerous visual elements need to be described, elements appearing in multiple segments can be omitted temporarily.

Challenge #2. It's crucial to determine the specific overlaps between consecutive ADs and decide during the curation process which details can be omitted and which are indispensable. However, identifying which information is repeated across consecutive ADs in a narrative sequence is complex. Moreover, it is also challenging to naturally remove specific information from a single sentence where multiple pieces of information are organically combined.

4. Design

Our goal is to generate AD that captures the narrative context of movies with an appropriate syllable count. As depicted in Fig. 2, NarrAD consists of three stages:

1. Video-to-script retrieval (§4.1). To obtain the narrative context necessary for AD generation, we identify which parts of the movie script correspond to the given video segment. To address inconsistencies, the identification is made at the level *scene*, the smallest narrative unit where the action remains continuous in one place. Unlike minor details at the sentence level, the overall temporal and spatial context at the *scene* level generally aligns, making it more robust to inconsistencies. Additionally, we enhance the accuracy by employing a two-fold approach that utilizes both dialogue and visual clues. (1) By aligning the movie's dialogue with the script, we roughly segment the movie, (2) then classify the given video segment according to *scene headings* for more precise extraction.

2. AD generation with a multimodal LLM (§4.2). Next, we generate AD by feeding a multimodal LLM with an appropriate instruction prompt, the narrative context, and the video segment. In this stage, we generate detailed ADs without considering the length of the narration slots.

3. Curating information across narration slots (§4.3). Finally, we curate the information from the generated text so that the ADs fit in the given narration slot while minimizing information loss. To address the issue of partial information overlap and naturally removing that information within a sentence, we adapt a disassembly-and-reassembly approach. We split sentences into minimal semantic units [36]. We compare these units with those from adjacent ADs to check if they have the same contents. We

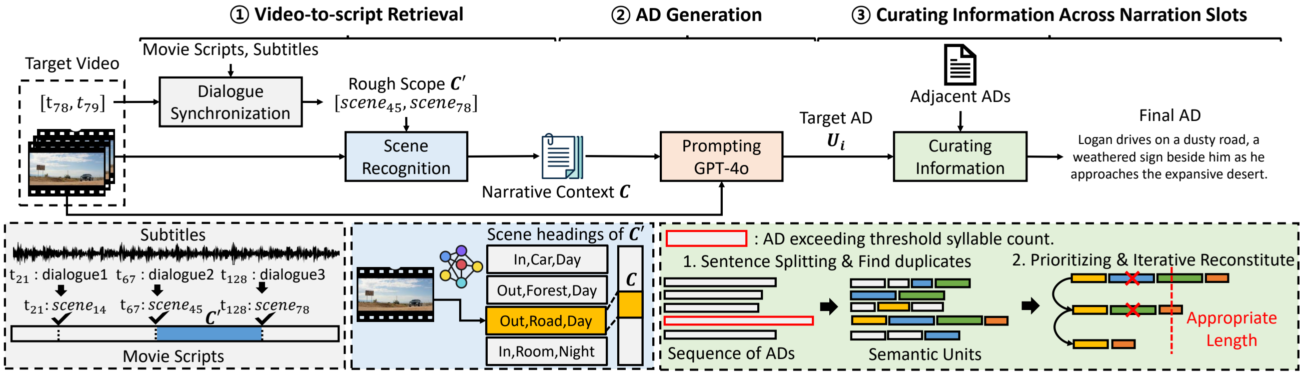


Figure 2. Overall design of NarrAD.

prioritize the units based on their uniqueness and sequentially curate them until we reach the desired syllable count.

4.1. Video-to-script Retrieval

The aim of video-to-script retrieval is to precisely extract narrative context relevant to the given video segment from the entire movie script at the *scene* level. A movie script S is divided into scenes based on scene headings. For a given timestamp $[t_i^{start}, t_i^{end}]$ and its corresponding video segment v_i, we identify the set of candidate scenes $C = \{s_1, s_2, \ldots\}$ to which v_i belongs. We obtain movie scripts from *IMSDb* and *The Script Lab*. The availability of scripts is evident in many works that utilize them in various contexts [6, 10–12, 22, 30, 31, 34, 35, 37–39, 43, 45].

Dialogue synchronization. We begin by narrowing down the search space by segmenting the movie script into rough sections, using dialogues as the first cue. By comparing the dialogue from the actual movie with the dialogue in the movie script, we create scene timestamps T, indicating which scene number in the script corresponds to a specific moment t on the movie timeline. We capture the dialogues and their timestamp from the movie $D = \{(d_1, t_1), (d_2, t_2), \ldots\}$. We also extract dialogues from the movie script and their associated scenes, denoted as $D' = \{(d'_1, s_1), (d'_2, s_2), \ldots\}$. By comparing each dialogue d_i in D with those in D', we identify matching dialogues d'_j. When d_i and d'_j are a definite match, we log the pair (t_i, s_j) in T.

To compare dialogues from two different sources, we use the Levenshtein distance [23], a string metric for measuring differences between two sequences of letters. We calculate the distance between d_i and all dialogues in D', matching d_i with the most similar dialogue in D'. However, due to inconsistencies between the movie and its script, it's not possible to match every dialogue. In addition, short exclamations, simple greetings, and commonly used phrases may align with multiple parts of the script. To ensure high recall, we use a filtering algorithm that selects only definite matches. If the matching result (d_i, D') meets any of the following conditions, we do not include it in the scene timestamps T: (1) d_i comprises four words or fewer; (2) the minimum distance from d_i to any dialogue in D' exceeds a set threshold; (3) more than two dialogues in D' have a distance from d_i that is less than the threshold; or (4) the scene s_i is earlier than s_{prev}, the most recently recorded scene in T, or (5) the gap between them is larger than 50 scenes; Finally, we remove pairs marked as the same scene if their time difference exceeds the set threshold. Using the scene timestamp T, we can identify the rough candidate scenes C' where the video segment with timestamp $[t^{start}, t^{end}]$ may belong.

Scene recognition. To pinpoint the most relevant narrative context, we employ scene recognition to identify a scene in the movie script that closely resembles the given video segment. In sections where definitive dialogue is lacking or significant inconsistencies with the movie script are present, the rough candidate scenes C', identified through dialogue synchronization, can become excessively long. If the word count of stage directions in C' exceeds the threshold M, we refine C' into more precise candidate scenes, denoted as C.

Our strategy is to compare the video background with the script because a scene maintains a consistent background in general. We utilize *scene headings* in a movie script that specify the background at the beginning of each scene, which includes three pieces of information: whether the scene takes place in an enclosed space (INT) or outdoors (EXT), the location, and the time of day. We determine the three background elements of the given video segment and select the scene headings from C' that best match these elements. For scene recognition, we utilize GPT-4o. We transform scene headings into natural phrases that capture the three pieces of information. For instance, the scene heading "EXT. BACKYARD - MORNING" is transformed into "outside the backyard, in the morning." By feeding the frames from the video segment, v_i, and the list of scene headings from C' into GPT-4o, we identify all scene headings that match the background of v_i. To ensure high recall, when the word count of stage directions in the selected scenes falls below a threshold, M, we expand the selection by including adjacent scenes until their combined length reaches M. The stage directions in the extended candidate scenes C are utilized as a narrative context.

4.2. AD Generation with Multimodal LLM

We utilize a multimodal LLM to generate video descriptions that reflect the narrative context of movie scripts, capitalizing on exceptional performance demonstrated by other studies [28, 57]. Each scene's stage directions contain the complete narrative context required to understand the video segments within that scene. Therefore, we utilize the stage directions from the scenes in C as the context of a multimodal LLM. We combine the narrative context in C, video frames v_i, and AD examples, and feed these into GPT-4o with an appropriate instruction prompt. For in-context learning [47], AD examples are provided by selecting ADs from the MAD training set [44] that correspond to slots of the same length as the given narration slot. At this stage, to maximize the information conveyed within the constrained narration slots, our design choice is to generate sufficiently detailed descriptions and selectively shorten overly long sentences. While it is possible to generate sentences that match the narration slot length from the start, this approach—by not considering other ADs and choosing among many visual elements to describe—is no different from independent curation and leads to significant information loss. Details of our instruction prompts are available in Supplementary Material (§A1).

4.3. Curating Information across Narration Slots

The goal of information curation is to shorten ADs that are too long for the given narration slot while minimizing information loss. Naïvely summarizing each AD independently causes significant information loss, so we consider adjacent ADs to prioritize and curate information based on redundancy. Given a sequence of ADs and their respective narration slot lengths $AD = \{(U_1, l_1), (U_2, l_2), \ldots\}$, we generate a sequence of curated descriptions $AD' = \{(U'_1, l_1), (U'_2, l_2), \ldots\}$, ensuring that the number of syllables in U'_i do not exceed the threshold syllable count for l_i. The threshold syllable count for l_i is determined through statistical analysis of the training set of the MAD dataset [44]. To determine the threshold syllable count for a given narration slot l_i, we first collect all ADs whose length of narration slot is l_i from the training set. Then, we calculate the N-th percentile value of the collected ADs' syllable counts and set it as the threshold. In this work, we set N to 75, which is a hyperparameter that can be adjusted according to user preferences. Here, N serves as a factor that adjusts the relationship between the syllable count of a sentence and the time it takes to pronounce it. Our information curation consists of three NLP tasks—sentence splitting, finding duplicates, and reconstitution—all of which are carried out using GPT-4o with in-context learning.

Sentence splitting. In general, a single sentence contains multiple pieces of visual information, such as actions, emotions, appearance, time, and space. To precisely identify and remove repeated information with other sentences, we split each sentence into semantic units, each holding only one piece of information. A semantic unit represents an intermediate structure that is simpler and more regular, typically comprising of a basic subject-predicate-object structure [36]. For each description U_i, we divide it into semantic units $\{u_{i1}, u_{i2}, \ldots\}$.

Finding duplicates & Prioritizing units. To identify the priority of the information, we look for redundant information in adjacent ADs. For the target AD U_i, AD within x seconds before and after is defined as adjacent AD. For each semantic unit u_{ij} within U_i, we compare it with all units from adjacent ADs to determine whether they duplicate in their meanings.

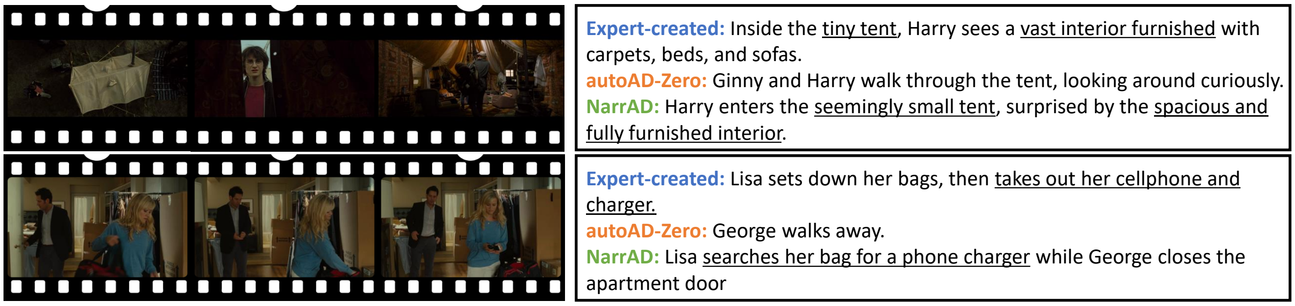
We establish the priority of a unit u_{ij} by evaluating how much its information is already conveyed by other units. The principle is straightforward: the more a unit's information is repeated or covered elsewhere, the lower the risk of losing the information it contains if it is not selected. Moreover, to ensure that no essential information is lost, at least one unit containing similar content must be preserved. When deciding which unit to retain, we rely on the normalized syllable count of the AD it belongs to. The normalized syllable count is defined as the current syllable count of the AD divided by the threshold syllable count for the given narration slot. If the AD has a higher normalized syllable count, more units need to be removed. To sum up, we prioritize units based on the number of duplicates with adjacent ADs that have a lower normalized syllable count than the current AD. The priority of a unit u_{ij} is expressed in a formula as follows:

$$\text{priority}(u_{ij}) = \sum_{u \in \text{adj}_{\text{low}}(U_i)} \text{duplicate}(u, u_{ij}) \tag{1}$$

$$\text{duplicate}(u, u_{ij}) = \begin{cases} -1, & \text{if } u \text{ and } u_{ij} \text{ have same information} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The $\text{adj}_{\text{low}}(U_i)$ represents the set of all semantic units from adjacent ADs of U_i, with the normalized syllable count lower than those of U_i. High priority indicates that the unit should be curated first.

Iterative reconstitution. Finally, we reconstitute the sentence by sequentially curating semantic units, starting with those of the highest priority. Until the sentence reaches the threshold syllable count, we repeat the curation-reconstitution process. Reconstitution involves creating new sentences by gathering information pieces from each semantic unit.



Expert-created: Inside the tiny tent, Harry sees a vast interior furnished with carpets, beds, and sofas.
autoAD-Zero: Ginny and Harry walk through the tent, looking around curiously.
NarrAD: Harry enters the seemingly small tent, surprised by the spacious and fully furnished interior.

Expert-created: Lisa sets down her bags, then takes out her cellphone and charger.
autoAD-Zero: George walks away.
NarrAD: Lisa searches her bag for a phone charger while George closes the apartment door

Figure 3. Qualitative results of NarrAD. We capture the narrative context of the scene as effectively as the expert-created [44].

Method	Training -free	Rou -geL	CIDEr	SPICE	R@5 /16
ClipCap [32]	X	8.5	4.4	1.1	36.5
autoAD [15]	X	11.9	14.3	4.4	42.1
autoAD2 [14]	X	13.4	19.5	-	50.8
autoAD3 [16]	X	-	24.0	-	52.8
VLog [2]	✓	7.5	1.3	2.1	42.1
MM-Vid [28]	✓	9.8	6.1	3.8	46.1
MM-Narrator [57]	✓	13.4	13.9	5.2	49.0
autoAD-Zero [52]	✓	14.4	22.4	7.3	46.5
NarrAD (Ours)	✓	**15.5**	**26.4**	**8.2**	**54.0**

Table 2. Comparison with prior works through captioning metrics.

	SegEval		LLM-AD-eval	
Method	Coherence	Specificity	L=1	L=5
autoAD-Zero [52]	0.49	0.50	0.93	1.12
NarrAD (Ours)	0.94	0.78	1.31	1.93

Table 3. Comparison with training-free works through LLM evaluator.

5. Evaluation

We evaluate NarrAD by employing standard captioning metrics and LLM-based metrics. In addition, we conducted a user study to evaluate the impact of NarrAD on real users. We evaluate NarrAD from the following perspective: (1) Does NarrAD generate higher quality ADs? (§5.1, §5.2) (2) Does information curation shorten original sentences to appropriate syllable counts while minimizing information loss? (§5.3) (3) In-depth analysis of the video-to-script retrieval module. (§5.4)

Dataset. Following the prior works [14–16, 28, 52, 57], we conduct experiments on the evaluation set of the MAD dataset [44] which is a collection of 6,520 expert-created ADs from 10 movies. Fig. 3 shows the ADs in comparison from an example scene.

Cost. NarrAD generates AD at a low cost. We used an average of 5.7K GPT-4o tokens, which cost $0.028 per AD. Considering that a typical movie includes approximately 685 ADs, the cost of generating AD for an entire movie is $19. This cost is significantly lower compared to the $1,800 per hour [24] required for human-produced content.

5.1. Evaluation using Natural Language Metrics

Method. We compare NarrAD to previous studies using reference-based captioning metrics, including RougeL [27], CIDEr [48], and SPICE [4], which measure the distance from the reference text. We also utilize the recall-based metric Recall@k within N neighbors (R@k/N), designed for text sequence generation [14]. In addition, many studies [9, 13, 16, 29, 57] show that LLMs have outstanding performance as evaluators for assessing generated text. Based on the idea, prior works [16, 57] propose new metrics for evaluating AD using LLMs. SegEval [57] groups consecutive ADs into a single segment and evaluates them at the sequence level in terms of coherence and specificity. It assesses how well the target segment performs in terms of coherence and specificity within the contextual window composed of ground truth AD segment. LLM-AD-eval [16] assesses the degree of alignment between the target AD and the reference AD, considering visual elements such as actions, objects, and interactions. Details of instruction prompts for LLM evaluators are available in Supplementary Material (§A2)

Comparison with prior works. Tab. 2 represents the comparative results of NarrAD with prior works through classic metrics for captioning. We first compare NarrAD with the training-free baselines [2, 28, 52, 57]. Like us, they generate ADs using task-specific prompts with off-the-shelf LLMs. However, the additional information they provide for AD generation, such as past ADs and character bank, is insufficient for comprehensive understanding the movie. NarrAD produces the highest quality ADs through the rich narrative context provided by the movie script. Moreover, we outperform even fine-tuning based works [14–16, 32] that have been trained on large amounts of data (CIDEr 26.4 vs 24.0). This demonstrates that by providing the LLM with appropriate additional information, it is possible to generate high-quality ADs even without the help of fine-tuning.

Comparison with SOTA using LLM. Tab. 3 provides a deeper evaluation of NarrAD and the training-free SOTA work, autoAD-Zero [52], using an LLM evaluator that is more specialized in assessing AD quality. Our detailed ADs with rich narrative achieve 1.92× higher Coherence

	The Ides of March				*How Do You Know*				*Charlie St. Cloud*			
Method	Use.	Spec.	Rec.	Comp.	Use.	Spec.	Rec.	Comp.	Use.	Spec.	Rec.	Comp.
Expert-created	$3.17_{\pm 1.7}$	$4.02_{\pm 1.8}$	$3.70_{\pm 2.0}$	$4.16_{\pm 2.2}$	$3.77_{\pm 1.6}$	$4.60_{\pm 1.6}$	$4.57_{\pm 1.9}$	$3.98_{\pm 1.7}$	$3.74_{\pm 1.6}$	$4.04_{\pm 1.7}$	$3.88_{\pm 1.8}$	$4.51_{\pm 1.6}$
MM-Narrator	$1.83_{\pm 1.6}$	$2.39_{\pm 1.8}$	$1.98_{\pm 1.5}$	$1.82_{\pm 1.7}$	$2.86_{\pm 1.5}$	$3.18_{\pm 1.8}$	$2.73_{\pm 1.8}$	$2.02_{\pm 1.5}$	$3.17_{\pm 1.6}$	$3.45_{\pm 1.8}$	$2.82_{\pm 1.9}$	$2.98_{\pm 1.6}$
autoAD-Zero	$1.59_{\pm 1.0}$	$1.56_{\pm 1.1}$	$1.48_{\pm 1.0}$	$1.60_{\pm 1.2}$	$2.99_{\pm 1.9}$	$3.02_{\pm 2.1}$	$2.96_{\pm 1.9}$	$2.27_{\pm 1.3}$	$2.98_{\pm 1.9}$	$3.00_{\pm 2.0}$	$3.24_{\pm 2.1}$	$3.54_{\pm 1.2}$
NarrAD (Ours)	$4.33_{\pm 1.5}$	$4.25_{\pm 1.6}$	$4.51_{\pm 1.8}$	$6.24_{\pm 1.1}$	$3.54_{\pm 1.4}$	$4.12_{\pm 1.7}$	$4.14_{\pm 2.0}$	$4.06_{\pm 1.2}$	$3.76_{\pm 1.6}$	$3.70_{\pm 1.9}$	$3.73_{\pm 1.8}$	$4.25_{\pm 1.4}$

Table 4. Results of survey and comprehension test. *Use.* represents usefulness, *Spec.* represents specificity, *Rec.* represents likelihood of recommendation, and *Comp.* represents score of the comprehension test. We recruit 200 participants per a movie.

and 1.56× Specificity compared to short, superficial, and repetitive ADs of autoAD-Zero. Furthermore, our ADs better align with the visual elements compared to the reference ADs, achieving 1.41× and 1.72× higher LLM-AD-eval score in text level (L=1) and sequence level (L=5) respectively. In the case of another baseline, MM-Narrator [57], although it generates detailed descriptions, the sentences are excessively long without considering the narration slot, making it difficult to function effectively as AD.

5.2. Evaluation with Human Subjects

We conduct a user study to evaluate the benefits of rich narrative context and narration slot-aware AD for actual users. Participants watch or listen to 90-second excerpts from a movie. They experience a total of two sets for each video: *Audio+AD*, *Video+AD*. They first listen to the *Audio+AD* set and take a comprehension test consisting of seven true/false statements about the movie's content. They answer each statement with "True", "False," or "I don't know." The number of correct answers is used to measure movie comprehension. In the second stage, participants watch the *Video+AD* set and carry out a survey to measure the quality of AD. We ask the participants to rate the following aspects of the AD on a scale from 1 to 7: (1) Usefulness: The AD aids in comprehending the movie's content, so the experience of listening to it is satisfying. (2) Specificity: The AD provides all the necessary information in sufficient detail. (3) Likelihood of recommendation: The AD is worth recommending to visually impaired people around me. We recruit 600 participants online and compare ADs obtained from the Expert-created [44], MM-Narrator [57], autoAD-Zero [52], and NarrAD.

Target movies. We selected three movies with different genres and varying proportions of dialogue for our study. An effective AD generation framework should provide AD tailored to various genres. Moreover, it should be capable of providing highly detailed AD to facilitate understanding when there is little dialogue, and delivering concise AD to avoid audio overlap when dialogue is abundant. To assess these criteria, we selected three movies: *The Ides of March* (political drama, dialogue-light), *How Do You Know* (romantic comedy, dialogue-heavy), and *Charlie St. Cloud* (fantasy, dialogue-light).

Results. Tab. 4 shows the scores for each survey item and the comprehension test. A multivariate analysis of variance (MANOVA) revealed a significant effect of AD type on all measures. In all three movies, NarrAD achieves significantly higher score across various aspects (p-values < .05) compared to autoAD-Zero. autoAD-Zero tends to focus mainly on the actions and expressions of characters, often missing critical contextual information necessary for understanding the storyline. Participants who listened to its AD frequently chose "I don't know", indicating that the AD did not explain the movie well. NarrAD also outperforms MM-Narrator in various aspects (p-values < .05). The low scores of MM-Narrator highlight the negative impact of audio overlapping on the user experience. Participants who listened to MM-Narrator reported feeling very uncomfortable due to the audio overlap. It generates detailed ADs, but the overlap reduces both the comprehension of the movie and the overall viewing experience. NarrAD shows performance that is either superior or comparable to expert-created AD. This suggests that the narrative context obtained from the movie script provides a level of satisfaction similar to that of expert-created content. Further details regarding the statistical analysis and experimental procedure are provided in the Supplementary Material (§A3).

5.3. Information Curation across Narration Slots

NarrAD adjusts the syllable count of the AD according to the given narration slot to minimize the audio overlap, while minimizing information loss.

Audio overlap is significantly reduced. Fig. 4 presents the results of measuring how much the ADs overlap with subtitles or other ADs. As we mentioned in §2, the time required to pronounce an AD with a specific syllable count is determined through statistical analysis of the training set of the MAD dataset [44]. We collect all ADs whose length of narration slot is l from the training set. Then, we determine the N-th percentile syllable count of those ADs as the appropriate syllable count for the length l. NarrAD reduces audio overlap significantly compared to NarrAD and MM-Narrator, regardless of how the N is set. When N is set as 75-tile, as we set during information curating, the average audio overlap time of NarrAD is 32.3% less than NarrAD without curation and 67.9% less than MM-Narrator.

AD length is proportional to the slot length. Fig. 5 shows the difference in the median number of syllables

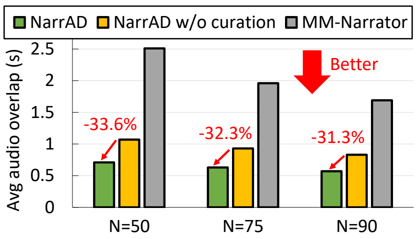


Figure 4. The average time of audio overlap for different pronunciation time factor N.

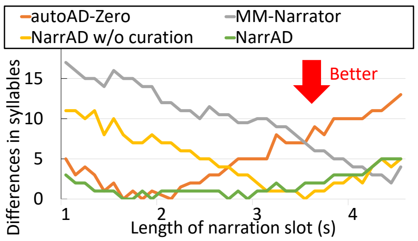


Figure 5. Difference in syllables from expert-created AD per narration slot.

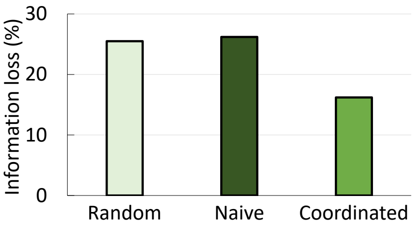
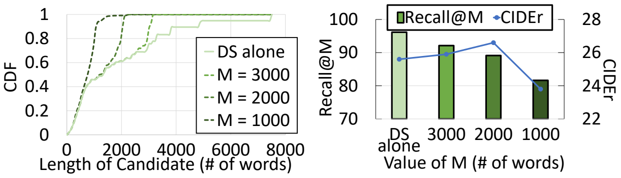


Figure 6. Information loss by curation methods.



(a) Length of candidate scenes

(b) Recall of candidate scenes.

Figure 7. The length and the recall of narrative context retrieved from movie scripts. The DS alone refers to the result of using dialogue synchronization alone.

between expert-created ADs from the MAD training set and others, across narration slots. Ideally, this relationship should be linear with the narration slots, as depicted in Fig. 1. However, both NarrAD without curation and MM-Narrator [57] show a significant difference from the expert-created ADs in most narration slots, as they consistently generate ADs that exceed the allowed length for the slot. As mentioned in §5.2, such long ADs hamper the movie experience. autoAD-Zero [52] also shows a large difference, as it always generates short ADs. Although its ADs do not overlap with other audio elements, they fail to adequately describe the movie because they convey a limited amount of information. In contrast, NarrAD's AD generation considers the length of the narration slot, generating concise ADs for short slots to avoid audio overlap, while generating detailed ADs for longer slots to provide a rich description.

Information loss. Fig. 6 represents the loss of information from three different curating methods. To quantify information loss, we measure the number of unique semantic units lost after curation. The number of unique semantic units is determined by counting the semantic units of entire ADs while excluding duplicates. The naïve approach curates sentences through the same GPT-4o, but does not take into account adjacent AD when prioritizing semantic units. The random approach is a method that arbitrarily assigns priorities to semantic units. Coordinated information curation (§4.3), which removes redundant information that overlaps with adjacent ADs, results in only 16.2% information loss. However, a naïve or random approach results in a much higher loss of information of 26.2% and 25.5%.

5.4. Analysis of Video-to-Script Retrieval

As we mentioned in §3, the shorter and more precise the movie script provided for narrative context, the higher the AD quality. We analyze the impact of the video-to-script retrieval module on AD quality by measuring the length and recall of candidate scenes extracted from the entire movie script using this module. Recall is defined as the proportion of samples in which the identified candidate scenes actually include a scene of video segment. We manually annotate each AD's corresponding scene, annotating 75.6% of the ADs that exist in movie scripts. Fig. 7a illustrates the length of narrative context retrieved from movie scripts. Fig. 7b shows the recall (bar) and AD quality (solid line) varies with the length threshold M of scene recognition. Using dialogue synchronization alone can achieve a high recall of 92.12, but the AD quality is relatively low (CIDEr 25.6) due to overly long length of the retrieved scenes. Therefore, we employ the scene recognition for more precise retrieval, balancing between length and recall based on the threshold M. As M increases, recall improves, but there is a trade-off in which the length of relevant scenes also increases, affecting the quality of AD. When M is set to 1k, 2k, and 3k, the recall gradually increases to 81.64, 89.13, and 92.12, respectively, but the length of the relevant scenes shorten. As a result of the trade-off effect, AD quality is highest (CIDEr 26.6) when M is set to 2k. Therefore, in all experiments, we configure M to be 2k.

6. Conclusion

We present NarrAD, a multimodal AD generation framework that provides rich contextual information through concise descriptions. This is the first work to utilize a movie script for AD generation to capture the narrative context of a movie. By curating information across narration slots, we adjust the syllable count of ADs to fit pauses in dialogues while minimizing information loss, an aspect that prior works have not considered. We achieve the highest ratings in captioning metrics, LLM based metrics, and user study among all prior designs.

Acknowledgments. We thank Ilyung Cheong for his invaluable assistance in data collection and analysis. We also thank anonymous reviewers for providing helpful feedback and suggestions to improve our work. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) [No. RS-2024-00406715].

References

[1] American council of the blind. https://adp.acb.org/, 2023. Accessed: 2023-03-23. 1

[2] VLog: A Framework for Video Understanding with Large Language Models. https://github.com/showlab/VLog, 2023. 6

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022. 2

[4] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 2, 6

[5] Wayner Barrios, Mattia Soldan, Alberto Mario Ceballos-Arroyo, Fabian Caba Heilbron, and Bernard Ghanem. Localizing moments in long video via multimodal guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13667–13678, 2023. 3

[6] Digbalay Bose, Rajat Hebbar, Krishna Somandepalli, Haoyang Zhang, Yin Cui, Kree Cole-McLaughlin, Huisheng Wang, and Shrikanth Narayanan. Movieclip: Visual scene recognition in movies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2083–2092, 2023. 2, 4

[7] Sihan Chen, Xingjian He, Longteng Guo, Xinxin Zhu, Weining Wang, Jinhui Tang, and Jing Liu. Valor: Vision-audio-language omni-perception pretraining model and dataset. *arXiv preprint arXiv:2304.08345*, 2023. 2

[8] Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Mingzhen Sun, Xinxin Zhu, and Jing Liu. Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. *arXiv preprint arXiv:2305.18500*, 2023. 2

[9] Cheng-Han Chiang and Hung-Yi Lee. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, 2023. 6

[10] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *2009 IEEE conference on computer vision and pattern recognition*, pages 919–926. IEEE, 2009. 2, 4

[11] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *2009 IEEE 12th International conference on computer vision*, pages 1491–1498. IEEE, 2009. 2, 3, 4

[12] Mark Everingham, Josef Sivic, and Andrew Zisserman. Hello! my name is... buffy”–automatic naming of characters in tv video. In *BMVC*, volume 2, page 6. Citeseer, 2006. 2, 3, 4

[13] Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, 2024. 6

[14] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD II: The Sequel - who, when, and what in movie audio description. In *ICCV*, 2023. 1, 2, 6

[15] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD: Movie description in context. In *CVPR*, 2023. 1, 2, 6

[16] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad iii: The prequel-back to the pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18164–18174, 2024. 1, 2, 6

[17] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6565–6574, 2023. 3

[18] Jinhyun Jang, Jungin Park, Jin Kim, Hyeongjun Kwon, and Kwanghoon Sohn. Knowing where to focus: Event-aware transformer for video grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13846–13856, 2023. 3

[19] Dohwan Ko, Ji Lee, Woo-Young Kang, Byungseok Roh, and Hyunwoo Kim. Large language models are temporal and causal reasoners for video question answering. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4300–4316, Singapore, Dec. 2023. Association for Computational Linguistics. 2

[20] Anna Kukleva, Makarand Tapaswi, and Ivan Laptev. Learning interactions and relationships between movie characters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'20)*, 2020. 2

[21] Weicheng Kuo, A. J. Piergiovanni, Dahun Kim, Xiyang Luo, Benjamin Caine, W. Li, Abhijit S. Ogale, Luowei Zhou, Andrew M. Dai, Zhifeng Chen, Claire Cui, and Anelia Angelova. Mammut: A simple architecture for joint learning for multimodal tasks. *ArXiv*, abs/2303.16839, 2023. 2

[22] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2, 3, 4

[23] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966. 4

[24] Elisa Lewis. How to select an audio description vendor, 2021. 1, 6

[25] Hongxiang Li, Meng Cao, Xuxin Cheng, Yaowei Li, Zhihong Zhu, and Yuexian Zou. G2l: Semantically aligned and

uniform video grounding via geodesic and game theory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12032–12042, 2023. 3

[26] Yi Li, Kyle Min, Subarna Tripathi, and Nuno Vasconcelos. Svitt: Temporal learning of sparse video-text transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18919–18929, 2023. 3

[27] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. 2, 6

[28] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. Mm-vid: Advancing video understanding with gpt-4v(ision). 2023. 2, 5, 6

[29] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, Dec. 2023. Association for Computational Linguistics. 6

[30] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2929–2936. IEEE, 2009. 2, 3, 4

[31] Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5257–5266, 2017. 2, 3, 4

[32] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 6

[33] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23023–23033, 2023. 3

[34] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10317–10326, 2020. 2, 4

[35] Iftekhar Naim, Abdullah Al Mamun, Young Chol Song, Jiebo Luo, Henry Kautz, and Daniel Gildea. Aligning movies with scripts by exploiting temporal ordering constraints. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1786–1791. IEEE, 2016. 2, 4

[36] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. Transforming complex sentences into a semantic hierarchy. *arXiv preprint arXiv:1906.01038*, 2019. 3, 5

[37] P. Papalampidi, F. Keller, L. Frermann, and M. Lapata. Screenplay summarization using latent narrative structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. 2, 4

[38] P. Papalampidi, F. Keller, and M. Lapata. Movie plot analysis via turning point identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019. 2, 4

[39] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie summarization via sparse graph construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13631–13639, 2021. 2, 4

[40] Renjing Pei, Jianzhuang Liu, Weimian Li, Bin Shao, Songcen Xu, Peng Dai, Juwei Lu, and Youliang Yan. Clipping: Distilling clip-based models with a student base for video-language retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18983–18992, 2023. 3

[41] Elisa Perego. Gains and losses of watching audio described films for sighted viewers. *Target*, 28(3):424–444, 2016. 1

[42] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212, 2015. 3

[43] Josef Sivic, Mark Everingham, and Andrew Zisserman. “who are you?”-learning person specific classifiers from video. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1145–1152. IEEE, 2009. 2, 4

[44] Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5026–5035, June 2022. 2, 3, 5, 6, 7

[45] Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. “knock! knock! who is it?” probabilistic person identification in tv-series. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2658–2665. IEEE, 2012. 2, 4

[46] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[47] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 200–212. Curran Associates, Inc., 2021. 2, 5

[48] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 2, 6

[49] Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In Houda Bouamor, Juan Pino, and Kalika

Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore, Dec. 2023. Association for Computational Linguistics. 2

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022. 2

[51] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10704–10713, 2023. 3

[52] Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad-zero: A training-free framework for zero-shot audio description. *arXiv preprint arXiv:2407.15850*, 2024. 1, 2, 6, 7, 8

[53] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Fei Huang, and Jingren Zhou. mplug-2: A modularized multi-modal foundation model across text, image and video. *ArXiv*, abs/2302.00402, 2023. 2

[54] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*, 2022. 2

[55] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, 2023. 2

[56] Youngjae Yu, Jiwan Chung, Heeseung Yun, Jongseok Kim, and Gunhee Kim. Transitional adaptation of pretrained models for visual storytelling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12658–12668, June 2021. 2

[57] Chaoyi Zhang, Kevin Lin, Zhengyuan Yang, Jianfeng Wang, Linjie Li, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Mm-narrator: Narrating long-form videos with multimodal in-context learning. 2024. 1, 2, 5, 6, 7, 8

[58] Yimeng Zhang, Xin Chen, Jinghan Jia, Sijia Liu, and Ke Ding. Text-visual prompting for efficient 2d temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14794–14804, 2023. 3

[59] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022. 2