REFRAMED: Towards Realistic Audio Description Generation for Movies

Igor Sterner, Mirella Lapata, Alex Lascarides & Frank Keller
School of Informatics
University of Edinburgh
United Kingdom
igor.sterner@ed.ac.uk, {mlap,alex,keller}@inf.ed.ac.uk
https://igorsterner.github.io/reframed/

Abstract

Audio Description (AD) is a verbal narration of key visual content in videos, enabling access for visually impaired audiences. Unlike standard video captioning, AD is a structured editorial task: descriptions must be inserted into gaps in dialogue and must convey only what is needed to understand the narrative being told. However, existing approaches formulate AD generation in an artificial setting where both the content and timing of descriptions are pre-specified, reducing the task to clip-level captioning. They further rely on noisy transcription and alignment pipelines, and lack the rich parallel data required for modeling narrative context. We introduce a new formulation of AD generation in which models must jointly decide *what* to describe and *when* to do it. To support this, we present **REFRAMED**, a high-quality dataset of 2,023 videos that span 3,302 scenes from 206 movies, with professional AD transcripts (both American and British versions), professional subtitles and aligned screenplays. We also provide a manually curated challenge set that pairs full movies with multiple AD references, together with evaluation protocols that leverage dialogue gaps and multi-reference comparisons. Experiments with state-of-the-art AD systems and multimodal LLMs show that they outperform trivial baselines but fall far short of expert human performance. Our dataset and benchmark establish a new foundation for research on video understanding.

1 Introduction

Audio Description (AD) is an accessibility service for visually impaired individuals. It is a verbal narration of key visual information necessary for understanding a video. The narration is timed to fit into natural gaps in dialogue, so that it minimally interrupts the original soundtrack. It is recorded and delivered via an earpiece or a secondary audio channel. Both American and British legislation will soon mandate AD at scale: Title II of the Americans with Disabilities Act will require public entities to provide AD for pre-recorded video, while the UK's Media Act sets a streaming quota of 10% by 2030. These mandates create an urgent need for automating AD in a way that addresses real-world requirements. However, current systems have only tackled limited aspects of the task.

For movies, the full task of creating AD requires a describer to first understand the movie's narrative arc before deciding which visual elements are needed to be able to follow the story. The describer must then formulate concise and evocative language that fits within dialogue gaps. Consider the AD in Figure 1, which is for the final scene in the 2011 murder-mystery *The Girl with the Dragon Tattoo*.[1] This excerpt shows that the creation of AD requires fine-grained visual understanding, character tracking, location and object tracking, conveying location changes, time, and mood. The AD also does not interrupt the dialogue.

[1]The video is available from second 37 onwards at https://rottentomatoes.com/m/the_girl_with_the_dragon_tattoo/videos/ofdvX951xl2S

Night. Lisbeth rides her motorcycle down a cobbled street with snow piled at the curbside. She parks on a corner outside Mikael's apartment and removes her crash helmet. She takes a package from the back of the bike, but stops dead when she sees Mikael leave his apartment with Erika.

[MIKAEL] We're late.
[ERIKA] Fashionably late.

She watches numbly as the couple walk away with their arms around each other. Lisbeth looks down as Mikael and Erika get into a waiting taxi. Lisbeth turns away and flings the package which has the card attached to it into a nearby dumpster. She puts her crash helmet back on, gets on her bike and starts it up. A solitary figure, she rides off into the night.

Figure 1: Excerpt of Audio Description (*italics*) and natural dialogue (**bold**) from *The Girl with the Dragon Tattoo* (2011). The AD requires fine-grained visual understanding, character and object tracking, and precise temporal placement, all without interrupting the soundtrack.

AD presents distinctive challenges that current systems are far from solving. In addition to fine-grained multimodal understanding over hours of content, AD generation requires precise control over when to describe and the ability to determine which elements of the movie are *narratively* salient. Such elements are not always *visually* salient: in the above scene, the two lovers are visible only from a distance (the middle frame of Figure 2), yet their actions are essential to the narrative and must be described.

Recent multimodal LLMs are candidates for meeting these challenges. They exhibit narrative understanding and can track characters, infer intent, and maintain coherence over long contexts, while grounding this reasoning in visual input. Moreover, the structured nature of AD, with fixed candidate gaps (between dialogue) and clear stylistic conventions, aligns naturally with the instruction-following capabilities of current models. Their ability to meet these challenges, however, cannot be meaningfully assessed as existing datasets and task formulations do not reflect the demands of AD in practice.

AD is created for nearly every major movie release, sometimes in multiple versions for a single movie, offering a rich source of expert-authored data that current datasets do not fully exploit. They all rely on unvalidated automatic speech recognition (ASR) to transcribe audio AD tracks, resulting in transcriptions we found to be extremely noisy. More fundamentally, prior work reduces AD generation to an artificial setting in which both AD *content-selection* and *placement* are fixed: models are given the clip that corresponds to the elements described in the reference AD. This collapses the problem to clip-level video captioning, ignoring the core editorial challenge of AD: deciding which elements to describe and when. Finally, existing datasets provide almost none of the rich parallel data (screenplays, professional dialogue subtitles, multiple AD versions) that would benefit novel treatments of the task.

To address these limitations, we reformulate AD generation as a joint decision problem over *when* and *what* to describe, and introduce **REFRAMED**, a dataset and benchmark for this setting. Our task formulation matches real-world requirements: given a video and gaps in dialogue, a model must determine where descriptions are needed and what to say. Uniquely, for both training and evaluation data we provide *two* reference English AD versions, screenplays aligned at the level of scenes, professional English dialogue subtitles and SDH subtitles, which include salient non-verbal audio for hard-of-hearing audiences. Moreover, we adapt existing evaluation metrics to our novel AD task and benchmark current AD generation systems and LLMs under this formulation. Our contributions are:

- **A realistic task formulation** for AD generation (a model must decide when and what to describe), together with evaluation metrics that exploit multiple references.
- **REFRAMED**, a dataset of 2,023 video excerpts (3,302 scenes) from 206 movies with dual AD versions (US and UK), dialogue subtitles, SDH, screenplays, and video.
- The **REFRAMED challenge set** of 10 fully manually annotated movies with 2–3 professionally transcribed AD versions each, scene boundaries, AD-to-scene labels, screenplay alignment, and professional subtitles.
- **A suite of benchmarks** for ASR quality, temporal alignment, screenplay alignment, scene segmentation, AD splitting, and AD generation on the above data.

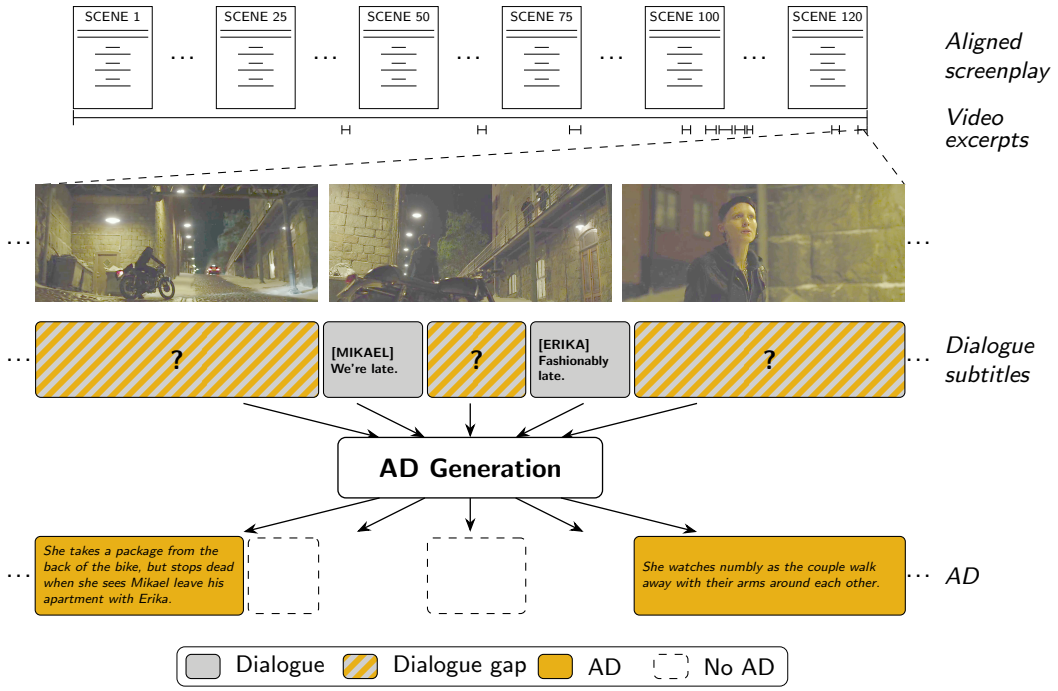


Figure 2: The **RE[FRAME]D** task formulation. Given input of a video, dialogue subtitles, and the spans of gaps between them, a model must decide both *when* and *what* to describe. Narrative context can be made available from the movie's screenplay and other scene videos.

2 Related Work

AD as Video Captioning. The task has commonly been framed as a form of video captioning: given a short video clip, generate a natural language sentence that describes its visually salient content. Most current work follows this formulation and finetunes multimodal models for the task, augmenting input with additional signals such as broader visual context, character information, or scene-level cues (Wang et al., 2021; Han et al., 2023a;b; 2024; Lin et al., 2024; Lee et al., 2025; Deganutti et al., 2025; Wang et al., 2025; Ye et al., 2025). DistinctAD (Fang et al., 2025) trains using character information, visual features and text from nearby clips, with architectural components and learning objectives that reward distinctive descriptions. Xie et al. (2024) propose to first generate a detailed description of a clip and then summarize it into AD; in Shot-by-Shot (Xie et al., 2025), this approach is improved by incorporating knowledge of shot boundaries and scale. Other work relies on LLM prompting (Lin et al., 2023; Chu et al., 2024; Zhang et al., 2024; Ye et al., 2024; Khandelwal et al., 2025); Gao et al. (2025) provide a broader survey of the area. A related line of work incorporates screenplays as narrative context. Screenplays have been estimated to contain approximately 60% of the information required for AD generation (Lakritz & Salway, 2006), and later work shows that screenplay context can improve generation quality (Rimle et al., 2020; Campos et al., 2023; Park et al., 2025). Despite this progress, all these approaches address a simplified formulation of AD creation: they assume AD placement is given and only generate the description sentence. Our benchmark is the first to require systems to determine placement and content jointly.

Movie Datasets with AD. Many movie datasets exist without AD (Tapaswi et al., 2016; Gu et al., 2018; Vicol et al., 2018; Huang et al., 2020; Bain et al., 2020; Wu & Krahenbuhl, 2021; Sun et al., 2025; 2024; Wu et al., 2024; Zaranis et al., 2025). The first large-scale dataset containing AD is LSMDC (Rohrbach et al., 2017), which pairs semi-automatically transcribed AD sentences with manually extracted movie clips that correspond with the described elements, replacing character names with placeholders. MAD (Soldan et al., 2022) scales this up by fully automatically transcribing AD tracks and aligning them to full movies

via audio cross-correlation. Evaluation material reuses LSMDC timecoding, which is suited to their task of video grounding. Han et al. (2023a) repurpose a subset of this evaluation data for a full-movie AD generation benchmark; the adoption of the LSMDC timecoding means that the time-span of the element to be described in the generation is provided as input. The placeholder substitution for character names is also retained. Han et al. (2024) introduce CMD-AD by aligning AD segments to movie scenes from YouTube and allowing temporal offsets and speed adjustments to account for differences in broadcast frame rates (e.g., 23.976fps NTSC vs. 25fps PAL).

These datasets share three limitations. First, AD transcripts are produced automatically, yet transcription quality has never been evaluated; in Appendix C.1 we show this is a major issue. Second, the alignment of AD tracks to movie scenes has also not been validated; in Appendix C.2 we show that the method used for the construction of CMD-AD is very noisy. Third, the MAD-based evaluation set inherits LSMDC timecoding, transcription errors and character name replacement, issues that have propagated into subsequent work.

Evaluation of AD Generation. Appendix Table 3 summarizes the many metrics used to evaluate AD generation. They include n-gram (e.g. METEOR, Banerjee & Lavie, 2005), image-captioning (e.g. CIDEr, Vedantam et al., 2015), and QA-based evaluation (Kala et al., 2025). Dense video captioning metrics such as SODA (Fujita et al., 2020) and its temporal counterpart (Batra et al., 2022) align generated and reference elements using their time spans before scoring them with F-measure. However, these metrics inherit limitations from the simplified task setting. Because prior work assumes fixed AD placement, evaluation operates at the level of individual captions matched to predefined clips and therefore cannot assess whether a system correctly decides *when* to insert a description as well as *what* visual element needs to be described. Prior work also ignores the task's subjectivity and typically uses only a single reference. We adapt existing metrics to our open-ended task formulation, and use multiple references throughout.

3 The REFRAMED Dataset

We now provide an overview of the REFRAMED dataset and its construction, including quantitative validation of each tool used (results are summarized in Section 3.2, with more details in Appendix C and E). Our design is guided by the goal of supporting AD generation as a joint decision problem over *when* and *what* to describe, informed by narrative context.

3.1 Data and Benchmark Overview

The dataset is built from movie excerpts licensed by Fandango, with raw videos without watermarking available publicly on `rottentomatoes.com`. Narrative context is provided from full-movie screenplays that we align to the movie timelines as well as other videos from the same movie. We select movies with at least five movie excerpt videos and listings on `audiovault.net` for *both* American and British AD versions. Table 1 compares our dataset against publicly available alternatives, showing our contributions of providing more parallel data streams for models to learn from, multiple AD references for both training and evaluation material, and gold standard data for evaluation. Following prior work (Bamman et al., 2024), we exclude animated movies as they represent a substantially different visual domain.

AD Transcription and Alignment. We obtain transcription for all AD tracks: for evaluation movies, they are the result of professional human transcription; training movies rely on a Speechmatics-based pipeline (Speechmatics are a specialist ASR and diarization provider). For each movie, we align the video excerpts to the full AD track (both American and British versions) using cross-correlation, accounting for possible PAL–NTSC speed differences and enforcing consistent alignment offsets across scenes from the same movie.

Parallel Data Streams. For each movie excerpt, we provide professional English dialogue-only and SDH subtitles sourced from `opensubtitles.org`. They are aligned to video using ASR-based timestamp matching (Appendix G). For the 85 movies with parsed screenplays available in MovieSum (Saxena & Keller, 2024), we align screenplay scenes to movie scenes

Name	Type	TRAINING							TESTING							TASK
		🎞	🔊	💬	📜	AD	hrs	#refs	🎞	🔊	💬	📜	AD	hrs	#refs	
LSMDC (AD)	5s clips	✔	✔	✔	✘	✔	134	1	✔	✔	✔	✘	✔	12	1	VC
MAD	full movies	✘	✘	✘	✘	✔	990	1	✘	✘	✘	✘	✔	217	1	VG
MAD-v2-eval	full movies	–	–	–	–	–	–	–	✘	✘	✔	✘	✔	19	1	VC
MAD-v3-eval	full movies	–	–	–	–	–	–	–	✘	✘	✔	✘	✔	19	1	ASR
REFRAMED	full movies	–	–	–	–	–	–	–	✔	✔	✔	✔	✔	21	2-3	AD
CMD-AD	2m scenes	✔	✔	✔	✘	✔	307	1	✔	✔	✔	✘	✔	22	1	VC
REFRAMED	2m scenes	✔	✔	✔	✔	✔	77	2	✔	✔	✔	✔	✔	4	2	AD

Table 1: Comparison of publicly available AD datasets. We report whether video (🎞), audio (🔊), subtitles (💬), screenplays (📜), and AD transcripts (AD) are available at gold-standard quality (✔), silver-standard quality (✔; transcription or alignment), or not at all (✘), alongside total hours of content and number of AD references. VC = video captioning; VG = video grounding; ASR = automatic speech recognition (and diarization) of AD; AD = our task of AD generation. Training includes validation data, if it is provided. CMD-AD hours are based on the available video data. Our contributions are highlighted.

using Vecalign (Thompson & Koehn, 2019) applied to joint dialogue and AD representations, enabling robust alignment despite script deviations.

AD Splitting and Scene Segmentation. We split AD transcripts into segments that each correspond to one visual element. For all evaluation data, the splitting is a part of the professional human transcription; for training data, it is from a learned splitting model. We additionally apply a second learned model that predicts scene boundaries from AD segments. Appendix K shows split elements for our earlier example.

Dataset Splits. The final dataset contains 206 movies and 2,023 video excerpts (average duration 144 seconds), spanning 3,302 scenes; 85 movies have aligned screenplays. We split the data into training, validation, and test sets such that the genre distribution is balanced across the splits (Appendix I). The training set excludes movies appearing in previous evaluation sets. The validation set consists of 10 movies from existing evaluation datasets, with aligned screenplays, while the test set contains 10 recent movies (2023–2025) not present in prior datasets. Validation and test movie excerpts benefit from professionally transcribed AD rather than automatic transcription (Appendix B).

Challenge Benchmark. For full-movie evaluation, we provide a manually annotated benchmark of 10 movies, each with 2–3 AD versions, exact scene boundaries, per-element scene labels, scene sequences, screenplay alignment, and professional dialogue and SDH subtitles (Appendix D). All annotations were performed manually, either by a professional post-production company or by a trained annotator. Seven of these movies are drawn from MAD-v3-eval; we additionally include *Dune: Part Two* (2024), *Barbie* (2023), and *Priscilla* (2023), selected for their award-winning AD tracks. The resulting set of ten movies spans 20 years of cinema and covers a diverse range of genres — 14 out of the 19 IMDb genre categories in the training set are represented.

3.2 The Construction Pipeline

Gold-standard Transcription. All downstream evaluation depends on gold-standard AD transcription, which is unavailable from existing data. We hence contracted a film post-production company to perform fully manual AD transcription and timecoding to the nearest frame (approximately 0.04 seconds), and to provide segments each corresponding to a single visual element of the movie, which we will call description elements. We call this new resource MAD-v3-eval. Analysis of these annotations shows that AD narration occurs within 10 seconds of the described elements, but not necessarily at exactly the same time as them, reflecting the need to place AD within available dialogue gaps (Figure 3).

AD Transcription and Diarization. The first pipeline stage takes a full-length AD audio track as input and produces a transcription with speaker labels and timestamps. We

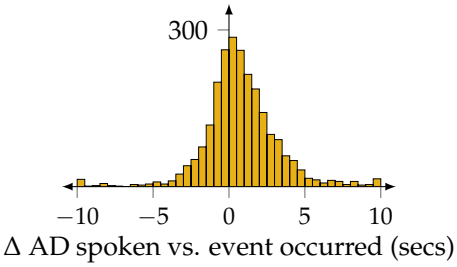


Figure 3: Histogram comparing (a) MAD-v2-eval timecoding (v2; when described elements occur) and (b) new professional timecoding (v3; when the AD is spoken). Texts are normalized and aligned at the word level, and v2 segments matching a concatenation of one or more v3 segments are retained. We plot the midpoint differences.

evaluate multiple ASR systems against MAD-v3-eval and find that existing AD corpora are extremely noisy (WhisperX achieves a best overall word error rate, WER=12.8; the MAD-v2-eval transcriptions achieve 22.1). A system based on Speechmatics with character-name adaptation achieves WER=2.9 and is used to transcribe all training data (Appendix C.1).

Temporal Alignment. The second stage aligns scene videos to full-length AD tracks using cross-correlation (Soldan et al., 2022), accounting for frame-rate differences between PAL and NTSC formats (Appendix Figure 5). This approach is robust for typical scene durations (100% alignment accuracy to within one frame for scenes of length 32 seconds or greater, which accounts for >99% of the video extracts in our data), and we enforce consistency of alignment offsets across scenes within each movie (Appendix C.2).

AD Splitting and Scene Segmentation. ASR produces sentence-level output, but our evaluation data is segmented into description elements. We therefore segment transcribed AD sentences using a RoBERTa-based model trained on gold-standard annotations (F_1=64.8 vs. a comma-splitting baseline F_1=36.9; Appendix E.3). A classifier over these segments then identifies scene boundaries, exploiting textual cues (e.g., "Now") that AD guidelines encourage describers to include (F_1=59.2; Appendix E.2). This produces scalable, silver-standard full-movie scene segmentation consistent across the AD versions. We combine these with scene boundaries derived from the start and end of each video excerpt (Appendix E.2).

Screenplay Alignment. We align screenplay scenes to movie scenes by representing dialogue and AD jointly as a screenplay-like sequence. We then apply Vecalign (Thompson & Koehn, 2019) to perform embedding-based dynamic programming alignment with support for scene insertions, deletions, and many-to-one merges (F_1=80.9 vs. a text-based dialogue-only baseline F_1=63.8; Appendix E.1). We apply this method to all movies with parsed screenplays available in MovieSum (85/206 movies; Saxena & Keller, 2024). This results in screenplay segments for 611 videos, after a filtering step that requires that every scene within a video successfully maps to the screenplay.

4 Evaluation Metrics

The constrained nature of AD lends itself well to evaluation, unlike many other long-form generation tasks. In particular, AD is anchored to dialogue gaps, description elements are typically inserted within a short temporal window of their occurrence (Figure 3), and multiple references are available in our data.

We evaluate AD generation under our task formulation, where models must jointly decide *what* visual information is narratively worth describing, *when* it can be inserted without interrupting dialogue, and *how* to express it concisely enough to fit the available time. These factors make no single evaluation metric sufficient. We therefore combine lexical similarity metrics, temporal alignment metrics, and QA-based semantic metrics. CIDEr and METEOR provide interpretable reference-based measures of content similarity. SODA-T evaluates whether generated descriptions are temporally aligned with professional AD references, using METEOR-based alignment, while QEval-T evaluates whether generated AD supports the same temporally grounded question answers as the reference. For evaluation on our challenge set, we additionally provide human-level results based on a third expert-created AD transcript.

Dialogue Gap-based Evaluation. Our first approach evaluates performance at the level of dialogue gaps, directly reflecting the main constraint that applies to AD placement. We collect gaps in the professional dialogue subtitles of at least 1 second, and assign references and generations to the gaps that fully contain them (allowing a 1 second gap collar). This assignment accounts for 90.2% of reference AD description elements in our challenge set: 87.8% (N=10,644) for American AD and 93.1% (N=8,734) for British AD, consistent with stricter British guidelines regarding AD placement. The remaining cases concern subtitled audio that evidently was not considered to preclude AD narration, such as musical lyrics or background dialogue. These remain covered by our non-dialogue gap metrics. Dialogue gaps that are not assigned any reference AD are discarded. Generated text within each gap is compared against the corresponding reference texts.

For the comparison, we compute CIDEr and METEOR scores. CIDEr natively supports multiple references. However, its length penalty makes it poorly suited to long references. For CIDEr, we therefore only retain dialogue gaps shorter than 20 seconds (see Appendix Table 18 for results on longer dialogue gaps). For METEOR, we follow prior work and take the maximum score across references. We retain all dialogue gaps for METEOR.

Alignment-based Evaluation. Because multiple nearby dialogue gaps may plausibly host some descriptions, strict gap-level matching is too rigid. SODA metrics (Section 2) allow us to relax the requirement that generated descriptions must align exactly with the reference gap. However, these metrics are unsuitable for direct application to AD, because they rely on the intersection of time spans to align generated and reference descriptions. Our adaptation of SODA is operationally similar to the original procedure: we compute an optimal one-to-one monotonic alignment between the reference and generated description elements using dynamic programming. The key difference in our approach is the alignment criterion. Our alignment optimizes only semantic similarity (METEOR F_1) between matched pairs. Sorted by start time, let us denote the sequence of n reference description elements as $\mathcal{D} = (d_i)_{i=1}^{n}$ and the sequence of $\hat{n}$ generated description elements as $\hat{\mathcal{D}} = (\hat{d}_j)_{j=1}^{\hat{n}}$. The alignment obtains $\mathcal{A}^\star = \arg\max_{\mathcal{A}} \sum_{(i,j)\in\mathcal{A}} \text{METEOR}(d_i, \hat{d}_j)$, computed exactly with dynamic programming (see Appendix L.1). We define SODA-M as the resulting F_1 measure over METEOR scores and SODA-T as the proportion of matched pairs with timespan midpoints differing by less than $\tau = 10$ seconds (chosen based on Figure 3; Appendix Figure 10 shows a sweep):

$$\text{SODA-M} = \frac{\sum_{(i,j)\in\mathcal{A}^\star} \text{METEOR}(d_i, \hat{d}_j)}{\frac{1}{2}\left(|\mathcal{D}| + |\hat{\mathcal{D}}|\right)}, \qquad \text{SODA-T} = \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{A}^\star} \mathbb{1}\left[|m(I_i) - m(\hat{I}_j)| < \tau\right]$$

where $m(I_i)$ is the midpoint of the time interval I_i of description element d_i. Following prior SODA implementations, we take the maximum of each score across references.

QA-based Evaluation. We assess whether generated AD supports downstream understanding using a QA-based evaluation that captures both semantic correctness and temporal grounding. We prompt OLMo 3 32B Think to generate multiple-choice questions from reference AD. Then, we filter aggressively for quality, retaining only questions answered correctly by OLMo 3 32B Base using each reference AD version and incorrectly when the supporting AD segment is removed. This ensures that questions depend on specific narrative content described in both AD references and that the wrong answer distractors are appropriate. Questions are answered using the base LLM by comparing multiple-choice logits. Prompts and more details on filtering are in Appendix L.2.

We define **QEval** and **QEval-T**, capturing semantic accuracy and temporally grounded accuracy. For each question, let y_j and $\hat{y}_j$ denote the gold-standard and predicted answers. I_j is given by the segment used to generate the question. We predict $\hat{I}_j$ as the segment whose removal most reduces the probability of the predicted answer. This attribution-based approach identifies which part of the generated AD supports the answer, and achieves round-trip consistency of 92.6% (see stages 3→4 in Appendix Table 13).

QEval measures answer correctness (the proportion of questions with system-predicted answer matching the gold-standard answer) and **QEval-T** also enforces temporal alignment:

$$\text{QEval} = \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}(\hat{y}_j = y_j)\ ;\ \text{QEval-T} = \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}(\hat{y}_j = y_j)\cdot \mathcal{C}_j\ ;\ \mathcal{C}_j = \mathbb{1}(|m(\hat{I}_j) - m(I_j)| < \tau)$$

where $\tau = 10$s (Figure 3). QEval-T therefore passes judgement on both the inclusion of specific narrative content (correct answer) and when it is grounded (correct temporal alignment). We retain only QA pairs that satisfy QEval-T under both reference AD versions; on average, there is one QA pair for every 3.6 reference description elements. Significance testing uses two-tailed Monte Carlo permutation tests with $R = 10{,}000$ and $\alpha = 0.05$.

5 Experiments

5.1 Experimental Setup

We demonstrate the application of our evaluation metrics to the **RE FRAME D** challenge set; results on the test set are in Appendix N. Our objectives are to assess: (1) **upper and lower bounds** on task performance; (2) progress made by **pre-existing models**; and (3) the extent to which **current LLMs** can perform our task.

Upper and Lower Bounds. As an approximate upper bound, we take a third professional AD transcript (available for one challenge movie) and evaluate it as a system output against the remaining two references. This gives a human-expert ceiling: i.e., the score a professional describer achieves on our metrics when judged against other professionals describing the same film. As a lower bound, we show results from a greedy baseline that fills each dialogue gap with randomly sampled training-set description elements until the gap is full and no further description elements can be added.

Pre-existing Modeling. Existing models were not designed for our task formulation (Section 2), so we give them the best possible chance by providing all information their designers made available, including gold-standard AD placement and content selection. For this evaluation, MAD-v2-eval timestamps are mapped to the new professional MAD-v3-eval timestamps, as is required for our evaluation (>99% of MAD-v2-eval segments receive mapped timestamps; the remaining are discarded). Seven of our ten challenge movies overlap with MAD-v2-eval, for which the required inputs are available; our measurements are an upper bound on these models' true performance.

Current LLMs. Many recent LLMs support up to one million tokens of context, and some are trained on multi-modal input, making full-movie input feasible. We evaluate two suitable model families. **Qwen 3.5** (Qwen Team, 2026) are open-weight models designed for video input and supporting context extension to one million tokens (via YaRN-based RoPE scaling, Peng et al., 2024); we run the 27B dense model. **Gemini 3.1 Flash-Lite** (Gemini Team, 2026) is a proprietary model at comparable per-token cost; we run it with `thinking=high` and default media resolution. All video input is at one frame per second with no audio. Qwen inference uses vLLM on two H200 GPUs; Gemini is accessed via the official paid API.

The models receive video, timed dialogue subtitles and a symbolic listing of dialogue gaps exceeding one second, and are instructed to generate zero or more timed description sentences per gap. In principle, they can take the full movie as input and output an AD script for it. However, we found that after generating approximately ten minutes of an AD script (c. 1k tokens), generation degenerates. For our experiments with full-movie input, we therefore instruct the models to generate descriptions for ten-minute chunks (specified via timestamps in the instructions). We compare this approach against providing each ten-minute video chunk in isolation, i.e., without the full movie as context. The generation process is iterative; generations for prior chunks are provided as a part of the input.

For our test set, where character names may not be inferable from the video alone, we provide manually annotated face crops of each character's first appearance alongside their name, in accordance with how a human describer would first assemble a character list (Appendix J). Prompts are in Appendix M.1.

	Expert	Random	D-AD	SbS	Qwen 3.5		Gemini 3.1	
					full	chunk	full	chunk
Dialogue-gaps								
CIDEr	**51.4**	1.3	15.6	16.3	10.3	13.2	12.2	19.0
METEOR	**20.1**	4.4	7.1	7.0	6.1	8.1	6.1	8.2
Aligned								
SODA-M	**16.3**	4.0	8.3	8.0	6.8	7.4	6.8	7.9
SODA-T	**81.0**	27.1	49.0	53.0	26.0	38.4	27.1	36.0
QA-based								
QEval	**69.6**	34.1	35.7	39.9	40.0	43.9	40.1	42.3
QEval-T	**61.2**	2.3	8.8	17.0	9.4	17.3	10.6	16.4

Table 2: Results on the **RE FRAME D** challenge set. *Dialogue-gaps*: comparisons within each dialogue gap. *Aligned*: scored textually (SODA-M) or temporally (SODA-T). *QA-based*: AD-generated answers scored against reference questions; QEval-T also penalizes descriptions at the wrong time. *full/chunk:* full movie vs. 10-minute video chunk as input.

5.2 Results

How well do humans agree? Table 2 summarizes results on our challenge set. Despite the subjectivity of AD, inter-human agreement is high. A QEval score of 69.6% shows that professional describers agree well on which elements of a movie are narratively salient, and a QEval-T of 61.2% confirms that they also describe them at similar times. N-gram agreement is likewise high (CIDEr = 51.4, METEOR = 20.1), validating our dialogue gap-based evaluation. SODA-M (16.3) is slightly lower than dialogue-gap METEOR, as expected given its stricter one-to-one alignment. SODA-T is high (81.0%) — even when human describers formulate AD differently, their descriptions overlap enough for temporal alignment.

Does the random baseline expose weaknesses in our metrics? The LLM evaluator answers 34.1% of questions correctly from random descriptions (QEval; chance-level=20%) demonstrating a degree of parametric knowledge and common-sense reasoning. A baseline with an empty context achieves 41.4% (all other metrics are zero), reinforcing this point. Nevertheless, random performance drops to 2.3% once temporal grounding is required (QEval-T). SODA-T reaches 27.1%: a result of monotonic alignment with random scores. All other metrics are near-floor, confirming that random descriptions receive no undeserved credit from our metrics.

How do pre-existing systems and LLMs perform? All differences between Shot-by-Shot (SbS) and DistinctAD (D-AD) against the random baseline are significant (for an evaluation of a wider range of systems, see Appendix Table 16). The largest gains are on CIDEr (DistinctAD: 15.6 and Shot-by-Shot: 16.3, vs. 1.3). SODA-T is trivially high, as expected, since these models are provided with gold-standard AD placements. The more recent Shot-by-Shot outperforms DistinctAD on both QA-based metrics (both $p < 0.001$).

The LLMs also outperform the random baseline. However, with full-movie input, they achieve significantly lower scores across all dialogue gap and aligned metrics compared to the pre-existing models. This reflects a difference in modeling approach: pre-existing models are provided with local context and generate plausible captions for each reference segment, whereas LLMs with full-movie input need to use timecoding to retrieve and describe the relevant part of their input. LLMs achieve the best results when provided with chunked input (ten minutes of context, as opposed to either a few seconds or the full movie), significantly outperforming the full-movie approach across all metrics. These results show that current LLMs cannot yet leverage full-movie narrative context, which is a requirement for realistic AD generation. Furthermore, AD requires precise temporal grounding, and our results show that this is currently less achievable with full-movie video input (SODA-T: Qwen 26.0% vs. chunk 38.4%, QEval-T: 9.4% vs. 17.3%).

[RON] He's supposed to be mad as a hatter, though, these days.

American AD: *The professors watch warily as Mad Eye Moody hobbles past. He scans the room with both eyes, then uses the bulging eye like a telescope to zoom in on Harry. Dumbledore shakes Moody's hand.*
British AD: *Moody has a scarred face and a large, protruding false eye fixed to a leather eye patch. It swivels erratically and lingers on Harry. One of Moody's feet is encased in a metal boot. Dumbledore...*

Qwen 3.5 (full-movie input): *The new professor shakes Dumbledore's hand while his multiple-lensed eye zooms in and out. He settles onto the bench next to Severus Snape, looking quite intimidating.*
Gemini 3.1 (full-movie input): *Filch stands alone in the aisles between the long tables, his hunched posture casting a gloomy presence over the Great Hall.*

[DUMBLEDORE] My dear old friend, thanks for coming.

Who does the swiveling eye linger on?

A. Harry B. Dumbledore C. Mr. Crouch D. Ron E. Moody

Figure 4: Two references and two model generated ADs based on a dialogue gap in the movie *Harry Potter and the Goblet of Fire* (2005). Dialogue in **bold**, AD in *italics*. The multiple-choice question is taken from our QA-based evaluation.

Consider Figure 4, which shows a twelve second dialogue gap twenty minutes into a *Harry Potter* movie. During this scene, two characters are introduced at a feast in a hall: Moody (a villain in disguise), and Dumbledore (a headmaster). Both LLMs are able to retrieve the relevant moment from their 2-hour video input and describe it with largely faithful details about a figure in a hall (note Qwen hallucinates that he sits down, and Gemini incorrectly names the character). However, they both fail to describe the narrative essence of the scene — Moody's appearance and his spying on Harry Potter foreshadow his status as a villain in disguise, as described by both references. The difference in length between the generations (21–26 words) and the references (34–35 words) is also notable. Professional AD in our challenge set is tightly centered around 200 words per minute, close to guideline recommendations. By contrast, Gemini generates descriptions that would need to be spoken much slower than the references (on average 110–118 words per minute; see Appendix Table 20). For the same example, the pre-existing AD systems generate more text (30–47 words), but their descriptions are less coherent and faithful (DistinctAD describes Moody as wearing a suit and sunglasses, with Ron and Harry standing in a forest).

What about the test set? Appendix Table 17 gives **RE FRAME D** test set results. Numbers are uniformly higher (Gemini: CIDEr 22.5, QEval-T 23.1%; Qwen: CIDEr: 18.0, QEval-T: 21.7%), suggesting this task is more tractable, though this gain stems from providing character names and faces: removing them drops CIDEr to 14.5 and 12.8 for Gemini and Qwen, respectively (both $p < 0.001$).

6 Conclusion and Future Work

We introduce a new formulation of AD generation that reflects real-world requirements: a model must decide both when and what to describe. To support this task, we introduce **RE FRAME D**, a dataset of 2,023 movie excerpts paired with multiple AD transcripts, dialogue subtitles, SDH, and, for 85 movies, aligned screenplays, together with a fully annotated benchmark of 10 full movies. We propose evaluation metrics that account for both the temporal constraints of AD and its inherent subjectivity. Validation against expert human AD shows high agreement. Prior AD systems and zero-shot LLMs outperform a random baseline but remain far below human performance.

Our results show that current models struggle with three coupled challenges. **Content selection** requires identifying narratively salient characters, objects, actions, and visual details across long contexts. **Temporal placement** requires finding suitable gaps within dialogue or salient audio. **AD realization** requires producing descriptions that are informative yet match the available narration time. **RE FRAME D** supports future work in these directions by providing dialogue subtitles, SDH subtitles, scene boundaries, aligned screenplays, and multiple professional AD references.

Ethics Statement

Providing equitable access to visual media requires AD to faithfully describe all key visual elements. This includes potentially sensitive topics such as sexual content, abuse and violence. This requirement places new pressure on the safety guardrails designed for current LLMs. In our evaluation of Gemini, one full movie (*Charlie St. Cloud (2010)*), two 10 minute chunks within the same movie, and two movie excerpts from *Speak No Evil (2024)* triggered non-configurable safety guardrails. This is likely to do with automatic detection of depictions of harm involving minors. For full-movie evaluation of Gemini, we report results based on the subset of movies without the one flagged. For our chunked and movie excerpt-based evaluation, we determined the output for these runs to be empty.

All AD transcripts for our evaluation data (validation, test and challenge splits) were the result of fully manual human transcription. We paid a post-production company contracted for the task their full rate for professional feature-movie subtitling.

In terms of data release, copyright is of potential concern here. The three main splits of our dataset contain data derived only from publicly available sources. We release the movie video excerpts as links and release all the other text-based scene data in full. The AD transcripts and dialogue subtitles are derived from excerpts of full-movie scripts, and our challenge set is based on full movies — we share these after researchers agree to terms of use. We release code necessary to evaluate future AD generation systems with the metrics proposed in this work.

Our dataset only represents English AD, since it forms the largest available data. In our companion paper (Sterner et al., 2026), we test hypotheses concerning the differences between American and British AD (for results separated by references, see Appendix Table 19). We expect many aspects of the formulation and evaluation setup to transfer cross-lingually. Professionally authored AD in many languages follows similar constraints regarding dialogue gaps and narrative accessibility. On the other hand, stylistic conventions vary across regions. Independently created AD tracks already exist in many languages for a subset of our movies, which could enable future cross-lingual extensions using our methodology.

Acknowledgements

Sterner is funded by an EPSRC PhD studentship (project reference 2923920). Lapata gratefully acknowledges the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1). We thank Arian Merati for his contributions to the evaluation of AD transcription and diarization. We are grateful to the team at Speechmatics for providing API access to their models and Google for providing Gemini API credits. We thank Inderjeet Mani, Argyrios Papoudakis, Olga Loginova, and three anonymous reviewers for their constructive feedback on this manuscript.

References

Apoorv Agarwal, Sriramkumar Balasubramanian, Jiehan Zheng, and Sarthak Dash. Parsing screenplays for extracting social networks from movies. In Anna Feldman, Anna Kazantseva, and Stan Szpakowicz (eds.), *Proceedings of the Workshop on Computational Linguistics for Literature (CLFL)*, pp. 50–58, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL https://doi.org/10.3115/v1/W14-0907.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 382–398. Springer, 2016. URL https://doi.org/10.1007/978-3-319-46454-1_24.

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed Movies: Story based retrieval with contextual embeddings. In *Proceedings of the Asian Conference on*

Computer Vision (ACCV), pp. 460–479, November 2020. URL https://doi.org/10.1007/978-3-030-69541-5_28.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. WhisperX: Time-accurate speech transcription of long-form audio. In *Interspeech*, pp. 4489–4493, 2023. URL https://doi.org/10.21437/Interspeech.2023-78.

David Bamman, Rachael Samberg, Richard Jean So, and Naitian Zhou. Measuring diversity in hollywood through the large-scale computational analysis of film. *Proceedings of the National Academy of Sciences (PNAS)*, 121(46):e2409770121, 2024. URL https://doi.org/10.1073/pnas.2409770121.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (eds.), *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.

Anil Batra, Shreyank N Gowda, Frank Keller, and Laura Sevilla-Lara. A closer look at temporal ordering in the segmentation of instructional videos. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2022. URL https://doi.org/10.48550/arXiv.2209.15501.

Giorgos Bouritsas, Petros Koutras, Athanasia Zlatintsi, and Petros Maragos. Multimodal visual concept learning with weakly supervised techniques. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4914–4923, 2018. URL https://doi.org/10.1109/CVPR.2018.00516.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://dl.acm.org/doi/abs/10.5555/3495724.3495883.

Virgínia P Campos, Luiz MG Gonçalves, Wesnydy L Ribeiro, Tiago MU Araújo, Thaís G Do Rego, Pedro HV Figueiredo, Suanny FS Vieira, Thiago FS Costa, Caio C Moraes, Alexandre CS Cruz, et al. Machine generation of audio description for blind and visually impaired people. *ACM Transactions on Accessible Computing*, 16(2):1–28, 2023. URL https://doi.org/10.1145/3590955.

Peng Chu, Jiang Wang, and Andre Abrantes. LLM-AD: Large language model based audio description system, 2024. URL https://doi.org/10.48550/arXiv.2405.00983.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. URL https://doi.org/10.1007/s11263-021-01531-2.

Adrienne Deganutti, Simon Hadfield, and Andrew Gilbert. DANTE-AD: Dual-vision attention network for long-term audio description. In *Proceedings of the Workshop on AI for Content Creation (AI4CC)*, 2025. URL https://doi.org/10.48550/arXiv.2503.24096.

Pelin Dogan, Boyang Li, Leonid Sigal, and Markus Gross. A neural multi-sequence alignment technique (NeuMATCH). In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8749–8758, June 2018. URL https://doi.org/10.1109/CVPR.2018.00912.

Mark Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy"–automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 2. Citeseer, 2006. URL http://doi.org/10.5244/C.20.92.

Bo Fang, Wenhao Wu, Qiangqiang Wu, Yuxin Song, and Antoni B. Chan. DistinctAD: Distinctive audio description generation in contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 13571–13581, June 2025. URL https://doi.org/10.1109/CVPR52734.2025.01267.

Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. Segment Any Text: A universal approach for robust, efficient and adaptable sentence segmentation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11908–11941, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://doi.org/10.18653/v1/2024.emnlp-main.665.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. Soda: Story oriented dense video captioning evaluation framework. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 517–531, 2020. URL https://doi.org/10.1007/978-3-030-58539-6_31.

Yingqiang Gao, Lukas Fischer, Alexa Lintner, and Sarah Ebling. Audio description generation in the era of LLMs and VLMs: A review of transferable generative AI technologies. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL*, pp. 471–490, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. URL https://doi.org/10.18653/v1/2025.findings-naacl.29.

Gemini Team. Gemini 3.1 Flash-Lite: Built for intelligence at scale, March 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-flash-lite/.

Sebastian Gil, Laney Kuenzel, and Caroline Suen. Extraction and analysis of character interaction networks from plays and movies. Technical report, Stanford University, 2011. URL https://snap.stanford.edu/class/cs224w-2011/proj/laneyk_Finalwriteup_v1.pdf.

Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6047–6056, 2018. URL https://doi.org/10.1109/CVPR.2018.00633.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. AutoAD: Movie description in context. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18930–18940, June 2023a. URL https://doi.org/10.1109/CVPR52729.2023.01815.

Tengda Han, Max Bain, Arsha Nagrani, Gul Varol, Weidi Xie, and Andrew Zisserman. AutoAD II: The sequel - who, when, and what in movie audio description. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 13645–13655, October 2023b. URL https://doi.org/10.1109/ICCV51070.2023.01255.

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18164–18174, June 2024. URL https://doi.org/10.1109/CVPR52733.2024.01720.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. MovieNet: A holistic dataset for movie understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 709–727, 2020. URL https://doi.org/10.1007/978-3-030-58548-8_41.

Divy Kala, Eshika Khandelwal, and Makarand Tapaswi. What you see is what you ask: Evaluating audio descriptions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 23496–23518, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. URL https://doi.org/10.18653/v1/2025.emnlp-main.1199.

Eshika Khandelwal, Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Andrew Zisserman, Gül Varol, and Makarand Tapaswi. More than a moment: Towards coherent sequences of audio descriptions, 2025. URL https://doi.org/10.48550/arXiv.2510.25440.

James Lakritz and Andrew Salway. The semi-automatic generation of audio description from screenplays. *Dept. of Computing Technical Report CS-06-05, University of Surrey*, 2006. URL https://andrewsalway.work/wp-content/uploads/2020/02/cs-06-05-1.pdf.

Anne Lambert, Marie Guégan, and Kai Zhou. Scene reordering in movie script alignment. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 213–218, 2013. URL https://doi.org/10.1109/CBMI.2013.6576585.

Seon-Ho Lee, Jue Wang, David Fan, Zhikang Zhang, Linda Liu, Xiang Hao, Vimal Bhat, and Xinyu (Arthur) Li. Now you see me: Context-aware automatic audio description. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, feb 2025. URL https://doi.org/10.1109/WACV61041.2025.00540.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

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. URL https://doi.org/10.48550/arXiv.2310.19773.

Kevin Qinghong Lin, Pengchuan Zhang, Difei Gao, Xide Xia, Joya Chen, Ziteng Gao, Jinheng Xie, Xuhong Xiao, and Mike Zheng Shou. Learning video context as interleaved multimodal sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 375–396, 2024. URL https://doi.org/10.1007/978-3-031-72967-6_21.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. In *Proceedings of the Conference on Language Modeling (COLM)*, 2024. URL https://doi.org/10.48550/arXiv.2401.17377.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL https://doi.org/10.48550/arXiv.1907.11692.

Yu Lu, Feiyue Ni, Haofan Wang, Xiaofeng Guo, Linchao Zhu, Zongxin Yang, Ruihua Song, Lele Cheng, and Yi Yang. Show me a video: A large-scale narrated video dataset for coherent story illustration. *IEEE Transactions on Multimedia*, 26:2456–2466, 2024. URL https://doi.org/10.1109/TMM.2023.3296944.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. URL https://doi.org/10.1109/ICCV.2019.00272.

Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. QuerYD: A video dataset with high-quality text and audio narrations. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2265–2269, 2021. URL https://doi.org/10.1109/icassp39728.2021.9414640.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie summarization via sparse graph construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 13631–13639, May 2021. URL https://doi.org/10.1609/aaai.v35i15.17607.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL https://doi.org/10.3115/1073083.1073135.

Jaehyeong Park, Juncheol Ye, Seungkook Lee, Hyun W. Ka, and Dongsu Han. NarrAD: Automatic generation of audio descriptions for movies with rich narrative context. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 409–419, February 2025. URL https://doi.org/10.1109/WACV61041.2025.00050.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://doi.org/10.48550/arXiv.2309.00071.

Charity Pitcher-Cooper, Manali Seth, Benjamin Kao, James M. Coughlan, and Ilmi Yoon. You Described, We Archived: A rich audio description dataset. *Journal on Technology and Persons with Disabilities*, 11:192–208, Jan 2024. URL https://pubmed.ncbi.nlm.nih.gov/38516032/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023. URL https://doi.org/10.48550/arXiv.2212.04356.

Philipp Rimle, Pelin Dogan-Schönberger, and Markus Gross. Enriching video captions with contextual text. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 5474–5481, 2020. URL https://doi.org/10.1109/ICPR48806.2021.9412008.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. URL https://doi.org/10.1109/CVPR.2015.7298940.

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)*, 123:94–120, 2017. URL https://doi.org/10.1007/s11263-016-0987-1.

Rohit Saxena and Frank Keller. MovieSum: An abstractive summarization dataset for movie screenplays. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL*, pp. 4043–4050, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://doi.org/10.18653/v1/2024.findings-acl.239.

Josef Sivic, Mark Everingham, and Andrew Zisserman. "Who are you?"-learning person specific classifiers from video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1145–1152, 2009. URL https://doi.org/10.1109/CVPR.2009.5206513.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba Heilbron, Chen Zhao, Silvio Giancola, and Bernard Ghanem. MAD: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5016–5025, June 2022. URL https://doi.org/10.1109/CVPR52688.2022.00497.

Igor Sterner, Alex Lascarides, and Frank Keller. Comparing British and American audio description of movies. In *Proceedings of the International Workshop on Computational Models of Narrative (CMN)*, June 2026. URL https://homepages.inf.ed.ac.uk/alex/papers/cmn2026.pdf.

Yidan Sun, Jianfei Yu, and Boyang Li. Multilingual synopses of movie narratives: A dataset for vision-language story understanding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP*, pp. 13488–13504, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://doi.org/10.18653/v1/2024.findings-emnlp.788.

Yidan Sun, Qin Chao, Yangfeng Ji, and Boyang Li. Synopses of movie narratives: a video-language dataset for story understanding. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2025. URL https://doi.org/10.1109/ICME59968.2025.11209961.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4631–4640, 2016. URL https://doi.org/10.1109/CVPR.2016.501.

Brian Thompson and Philipp Koehn. Vecalign: Improved sentence alignment in linear time and space. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1342–1348, Hong Kong, China, November 2019. Association for Computational Linguistics. URL https://doi.org/10.18653/v1/D19-1136.

Robert Turetsky and Nevenka Dimitrova. Screenplay alignment for closed-system speaker identification and analysis of feature films. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, volume 3, pp. 1659–1662. IEEE, 2004. URL https://doi.org/10.1109/ICME.2004.1394570.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015. URL https://doi.org/10.1109/CVPR.2015.7299087.

Subhashini Venugopalan, Amy Pavel, Tyler Roper, Jimmy Tobin, Emily Wilson, Jenny Tan, Alicia Martin, and Anton Kast. Metrics for fine-grained evaluation of inline audio descriptions. In *Proceedings of the Workshop on Vision Foundation Models and Generative AI for Accessibility: Challenges and Opportunities*, 2025. URL https://openreview.net/forum?id=HEzz8aNSW0.

Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. MovieGraphs: Towards understanding human-centric situations from videos. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8581–8590, 2018. URL https://doi.org/10.1109/CVPR.2018.00895.

Hanlin Wang, Zhan Tong, Kecheng Zheng, Yujun Shen, and Limin Wang. Contextual ad narration with interleaved multimodal sequence. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 8372–8383, June 2025. URL https://doi.org/10.1109/CVPR52734.2025.00784.

Yujia Wang, Wei Liang, Haikun Huang, Yongqi Zhang, Dingzeyu Li, and Lap-Fai Yu. Toward automatic audio description generation for accessible videos. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. URL https://doi.org/10.1145/3411764.3445347.

David Winer and R Young. Automated screenplay annotation for extracting storytelling knowledge. In *Proceedings of the Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, volume 13, pp. 273–280, 2017. URL https://doi.org/10.1609/aiide.v13i2.12994.

Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1884–1894, 2021. URL https://doi.org/10.1109/CVPR46437.2021.00192.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2024. URL https://dl.acm.org/doi/10.5555/3737916.3738823.

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. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 2265–2281, December 2024. URL https://doi.org/10.1007/978-981-96-0908-6_5.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Eshika Khandelwal, Gül Varol, Weidi Xie, and Andrew Zisserman. Shot-by-Shot: Film-grammar-aware training-free audio description generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 16503–16513, October 2025. URL https://doi.org/10.1109/ICCV51701.2025.01532.

Xiaojun Ye, Junhao Chen, Xiang Li, Haidong Xin, Chao Li, Sheng Zhou, and Jiajun Bu. MMAD: Multi-modal movie audio description. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pp. 11415–11428, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.998.

Xiaojun Ye, Chun Wang, Yiren Song, Sheng Zhou, Liangcheng Li, and Jiajun Bu. FocusedAD: Character-centric movie audio description, 2025. URL https://doi.org/10.48550/arXiv.2504.12157.

Zihao Yue, Qi Zhang, Anwen Hu, Liang Zhang, Ziheng Wang, and Qin Jin. Movie101: A new movie understanding benchmark. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 4669–4684, Toronto, Canada, July 2023. Association for Computational Linguistics. URL https://doi.org/10.18653/v1/2023.acl-long.257.

Zihao Yue, Yepeng Zhang, Ziheng Wang, and Qin Jin. Movie101v2: Improved movie narration benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 17081–17095, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://doi.org/10.18653/v1/2025.acl-long.836.

Emmanouil Zaranis, António Farinhas, Saul Santos, Beatriz Canaverde, Miguel Moura Ramos, Aditya K Surikuchi, André Viveiros, Baohao Liao, Elena Bueno-Benito, Nithin Sivakumaran, Pavlo Vasylenko, Shoubin Yu, Sonal Sannigrahi, Wafaa Mohammed, Ben Peters, Danae Sánchez Villegas, Elias Stengel-Eskin, Giuseppe Attanasio, Jaehong Yoon, Stella Frank, Alessandro Suglia, Chrysoula Zerva, Desmond Elliott, Mariella Dimiccoli, Mohit Bansal, Oswald Lanz, Raffaella Bernardi, Raquel Fernández, Sandro Pezzelle, Vlad Niculae, and André F. T. Martins. Movie facts and fibs (MF^2): A benchmark for long movie understanding, 2025. URL https://doi.org/10.48550/arXiv.2506.06275.

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. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13647–13657, June 2024. URL https://doi.org/10.1109/CVPR52733.2024.01295.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL https://doi.org/10.48550/arXiv.1904.09675.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models, 2025. URL https://doi.org/10.48550/arXiv.2506.05176.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, 2023. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/3666122.3668142.

A Other Datasets and Evaluation Metrics

Other Datasets. Other datasets also exist, such as TV-AD (Xie et al., 2024), YouDescribe (Pitcher-Cooper et al., 2024) (as used by Oncescu et al. (2021)) and Movie101 v1 and v2 (Yue et al., 2023; 2025). Datasets of movie summaries, also known as recaps, exist (Dogan et al., 2018; Sun et al., 2025; 2024; Lu et al., 2024). Our choice to work with AD for entire movies is based on the fact that they are self-contained and have long-form, complex narratives. Datasets of egocentric video with parallel descriptions are similar to AD in providing real-time narration of visual content. Epic Kitchens 100 (Damen et al., 2022) and HowTo100M (Miech et al., 2019) are two examples. These corpora are specifically designed to tackle action recognition, which is not equivalent to the task of generating AD.

Evaluation. Table 3 gives, to the best of our knowledge, an exhaustive summary of metrics that have been applied to the previous task of AD as video captioning. Rows (1)–(4) are variants of n-gram metrics, (5)–(6) were designed for image captioning using semantic scene parses or neural embeddings, (7)–(8) are tailored to report on repetition and character naming in AD, (9–11) use LLMs, either generation or encoder models, (12) addresses only verbs in generated AD, and (13)–(14) are an existing QA evaluation setup for AD.

	Metric	First proposed by	For AD proposed by	Description
(1)	BLEU	Papineni et al. (2002)	Rohrbach et al. (2015)	n-gram precision + brevity penalty
(2)	METEOR	Banerjee & Lavie (2005)	Rohrbach et al. (2017)	F_{mean} of unigram with stem and synonym matching + fragmentation penalty.
(3)	ROUGE-L	Lin (2004)	Han et al. (2023a)	F_1 of longest common subsequence
(4)	CIDEr	Vedantam et al. (2015)	Han et al. (2023a)	Cosine similarity of TF-IDF on n-grams
(5)	SPICE	Anderson et al. (2016)	Han et al. (2023a)	F_1 of a semantic graph
(6)	BERT-Score	Zhang et al. (2020)	Han et al. (2023a)	F_1 of BERT embeddings
(7)	Recall@k/N	—	Han et al. (2023b)	Rank window-N references by BERT-score and check if the gold pair is in top-k
(8)	Character names	—	Han et al. (2024)	Proportion of reference characters mentioned
(9)	LLM-as-a-Judge	Zheng et al. (2023)	Han et al. (2024)	Prompt an LLM for a score in [0, 5]
(10)	LLM-as-a-Judge (via AD guidelines)		Venugopalan et al. (2025)	Prompt an LLM for scores according to criteria in AD guidelines
(11)	Sentence similarity		Xie et al. (2025)	Max sentence similarity in a window
(12)	Verb lemmas		Xie et al. (2025)	Proportion of reference verbs mentioned
(13)	Fact-based QA		Kala et al. (2025)	Multiple-choice QA using LLMs + correct rationale
(14)	Narrative-based QA		Kala et al. (2025)	As above, but based on aligned plot summary sentences

Table 3: Evaluation metrics for AD generation. To the best of our knowledge, an exhaustive list of metrics previously applied to AD (framed as video captioning), grouped by type: n-gram overlap (1–4), image-captioning (5–6), AD-specific repetition and character naming (7–8), LLM-based (9–11), verb coverage (12), and QA-based (13–14). *First proposed by* gives each metric's origin; *For AD proposed by* gives the first work to apply it to AD.

B Professional AD Transcription

For AD transcription in the evaluation sets of **REFRAMED** and for our use for evaluating data collection techniques, we contracted the work of a movie post-production company. They performed fully manual AD transcription and timecoding to the nearest frame (approximately 0.04 seconds). The delivery consists of segments of transcription, each with a start and end timecode. One segment was defined as a single event, action or other visual element being described (they are frequently sentence-fragments).

We chose a company with experience with feature-film subtitling and AD, and one that assured us that they do not use any automatic AI tools in their work. For each movie that we provided them, the company researches the movie and the spellings of particularly hard characters and movie-related words. The timestamps are manually determined based on clicking between individual frames in the video and listening to when the audio begins. The data went through a full round of professional quality control from the company.

C Evaluating Pre-existing Data Collection Techniques

Here we evaluate the effectiveness of pre-existing data collection techniques. The first task is for a system to take as input a movie-length AD track, and output a transcription of it (text of what was said) and diarization (speaker labels for each span within the transcription; needed to differentiate character dialogue from the AD narrator). The second task is for a system to take as input a short movie clip and a full-length AD movie track, and output the offset and speedup of the clip with respect to the full track.

In order to define the first task, we need a gold-standard AD transcription. For the second task, we need aligned full-length movie and AD tracks, and then we can create evaluation data by extracting segments from the movie track and applying speed changes. For these evaluations, we used a newly transcribed version of MAD-v2-eval, which we call MAD-v3-eval. We purchased the 10 movies in this evaluation set. Figure 3 on page 6 shows the differences between MAD-v2-eval and MAD-v3-eval timestamps. A positive difference indicates that the AD described the event before it occurred, and vice-versa. A Gaussian best fit shows that on average AD describes events 0.8 seconds in advance, and the standard deviation is 2.4 seconds. This spread demonstrates that a central part of the task of creating AD is identifying suitable gaps in which to place a description that is nearby to its corresponding visual content; the visual event being described is not necessarily at exactly at the same time as when it is described in the AD.

C.1 Evaluating Automatic AD Transcription.

We compare the capabilities of three ASR systems on the task of transcribing AD: WhisperX (Bain et al., 2023) as used for the CMD-AD data; industry APIs from Microsoft (as used for the MAD training data); and Speechmatics (a reputable company that specialises in ASR). We can also evaluate the quality of the MAD-v2-eval transcripts against our new reference. Characters play a central role in movie narratives, and hence it is vital that they are transcribed correctly. We apply pre-existing methods for guiding ASR models to spell particular words correctly, which for us is character names. For more details, see Section F.

Results are in Table 4. MAD-v2-eval achieves overall WER of 22.1%, largely as a result of a high deletion rate: evidence that this data is of insufficient quality. WhisperX is better with a WER of 12.8%. Across all metrics, the Speechmatics systems perform best with overall WER of 2.9%. The good overall performance can be traced back, in part, to excellent diarization performance. They also provide the only systems with $\leq$1% timestamp error rate within 0.2 seconds. With our character adaptation, it spells $\leq$5% of character names incorrectly.

C.2 Evaluating Temporal Alignment of Scenes to Whole Movies.

The task now is to temporally identify where in a full-length AD track a given movie scene came from. This task is made more challenging in practice because of the speed differences

	ASR				Diarization			Names	Times		Overall			
	SR	DR	IR	WER	DER	MR	FAR	Micro	0.2	0.4	SR	DR	IR	WER
MAD-v2-eval											4.6	16.7	0.8	22.1
Microsoft	4.5	1.2	1.8	7.5	20.6	0.4	20.2	25.9	2.8	0.3	4.6	0.6	12.6	17.8
WhisperX	2.9	2.7	1.0	6.6	14.9	3.7	11.3	22.3	5.1	1.0	3.6	2.4	7.7	13.7
Speechmatics	**2.3**	**0.7**	**0.6**	**3.6**	**2.5**	**0.3**	**2.2**	**16.2**	**0.7**	**0.0**	**2.3**	**0.5**	**0.8**	**3.6**
Character-name adapted														
WhisperX	1.7	2.6	1.1	5.4	14.1	3.4	10.7	7.2	5.3	1.0	2.2	2.5	8.1	12.8
Speechmatics	**1.5**	**0.7**	**0.6**	**2.8**	**2.7**	**0.2**	**2.5**	**4.8**	**0.8**	**0.1**	**1.5**	**0.5**	**0.8**	**2.9**

Table 4: Automatic AD transcription error rates (%); lower is better, best per column in **bold**. **ASR** (with gold-standard diarization): word error rate (WER) and its substitution (SR), deletion (DR), and insertion (IR) components. **Diarization**: binary diarization error rate (DER), with miss rate (MR) and false alarm rate (FAR). **Names**: character-name error rate, micro-averaged over reference characters. **Times**: timestamp error rate within a collar of 0.2 or 0.4 seconds. **Overall**: end-to-end WER (and SR/DR/IR) after automatic diarization. The lower block repeats the systems with character-name adaptation. See Appendix F and Tables 11–12 for full details and all system variants.

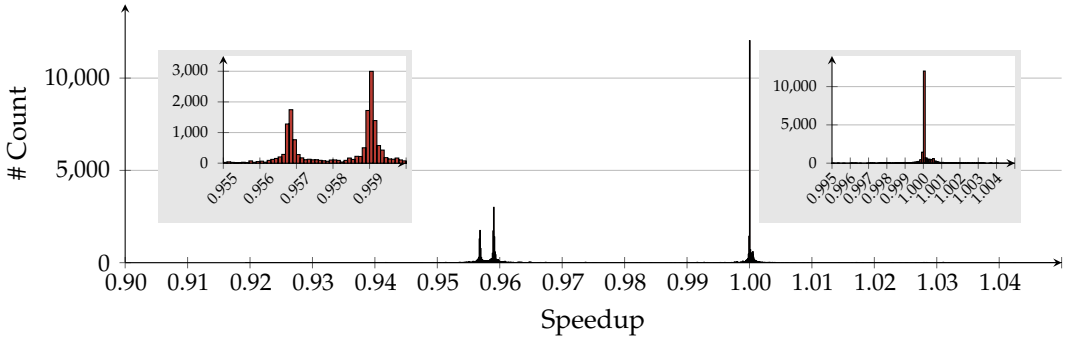


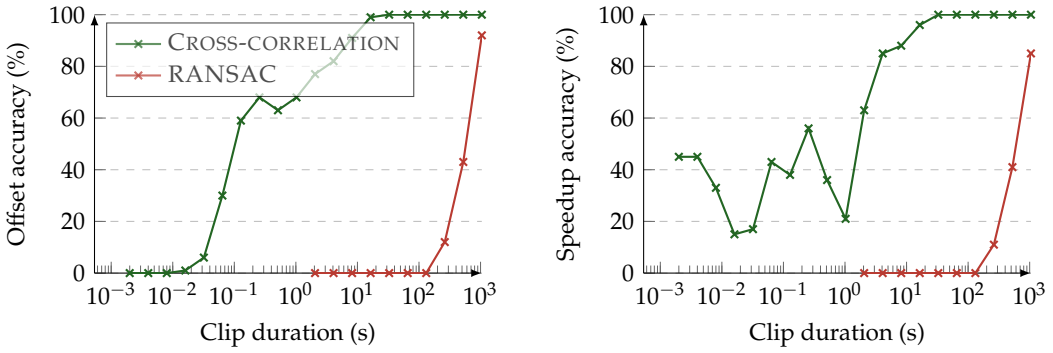
Figure 5: A histogram of the speedups computed by Han et al. (2024) for CMD-AD.

caused by the different frame rates of the two main standards: NTSC (as used in North America, which uses 24000/1001 frames per second) and PAL (as used in Europe, which uses 25 frames per second). In order to treat potential speed differences, all we need to do is allow for one version to be NTSC and the other PAL, or both are NTSC/PAL. Figure 5 shows reverse-engineered data released by Han et al. (2024) — experimentally we find that their predicted speedups are consistent with these two theoretical results.

Hence our task definition is: given a full-length AD track and a movie scene, identify the time of the start of the movie scene in the AD track and determine if the scene needs a PAL-NTSC speeddown, no speed change, or a NTSC-PAL speedup. We create many samples of each case by using the pre-aligned full movies in MAD-v3-eval, and we do so for many different lengths of movie scene. Han et al.'s (2024) RANSAC-based method can directly be applied to the task; for evaluation, we consider any predicted speed difference within 0.01 of the reference to be correct. Soldan et al.'s (2022) cross-correlation can only predict the offset, so we adapt it by repeating the cross-correlation computation for each of the possible speed changes and use the one with highest correlation score to be the prediction. We consider offsets to be correct if they are within 0.04 seconds

Parameter	Value
Target sample rate	16000
Number of Mel bands	128
FFT window size	1024
Hop length	512
RANSAC window size	50

Table 5: Hyperparameters for audio feature extraction and cross-correlation alignment.



(a) Accuracy of determining movie scene offset. (b) Accuracy of determining the speed difference.

Figure 6: Scene-to-movie temporal alignment accuracy as a function of scene duration, comparing cross-correlation (Soldan et al., 2022) against the RANSAC-based method of Han et al. (2024). Cross-correlation reaches perfect accuracy for scenes of $\geq$32 seconds, while the RANSAC method is reliable only at much longer durations; we therefore adopt cross-correlation in **REFRAMED**.

of the reference. Table 5 details the specific hyperparameters used for feature extraction and alignment.

Figure 6 provides results. We can see that the success of the methods depends greatly on the length of the scene to be aligned. The only approach to achieve perfect performance is using cross-correlation on scenes of at least 32 seconds. The RANSAC-based method performs poorly without very long scene durations. Our experiment here also suggests why the RANSAC method is inferior—it relies at its core on applying many cross-correlation to 3 second windows, and we have shown here that cross-correlation is very noisy at this time-scale.

In **REFRAMED**, we opt to use cross-correlation. In order to guarantee the highest possible quality assurance, we additionally require that all the offsets for a video from a given movie impute an OLS regression line between the US and UK AD tracks with RMSE ≤ 1. We state that we have high confidence in the alignment of the resulting videos based on the fact that it would be extremely unlikely for $\geq$5 independent alignment offsets to maintain linear consistency across the entire movie if one or more was erroneous.

D Challenge Set Annotation

The data in the **REFRAMED** challenge set of full movies serves both as an evaluation for AD generation and as an evaluation material for further data collection steps that we perform. Annotation for this set was performed manually.

High-level Overview. There are 10 movies, and each movie is provided with 2-3 AD transcripts. The movies are either 2023/2024 movies with award winning AD tracks (2/10), have three independently-created AD versions (1/10), or are the same movies as in MAD-v3-eval (7/10). The exact frame that starts each new scene in every movie is labeled. Each AD segment is labeled with a label of the scene whose content it describes. The screenplays scenes are mapped to the movie scenes. Boundaries between sequences of scenes are obtained from the official movie release chapter boundaries. English SDH and English dialogue subtitles are from the official movie release.

Details. For each movie in our benchmark, we require the availability of independently created US and UK AD versions for use as multiple references. For seven out of the ten movies in MAD-v3-eval, we were able to identify and purchase movies with the other AD

version to the version in the previous data. We add two movies which received awards for the quality of their AD tracks in the AD Awards gala and/or EGA Hermes Awards and also have US and UK AD versions, namely Dune Part Two (2024) and Barbie (2023). We add a final movie for which we were able to obtain three independent AD tracks (two US versions, one UK version), namely Priscilla (2023). The benchmark now contains movies that span two decades (2005–2024) and a wide range of genres.

We contracted the film post-production company to provide us with professional AD transcripts for each of the 21 (9 movies with two references, 1 movie with three references) tracks. An annotator manually determined if PAL–NTSC speedups and/or offsets are required between the versions and ensure that all versions are temporally aligned, determined the scene boundaries in the movie and determined which AD segments correspond to which scene. The process was as follows: (1) using professional subtitling software, we labeled the exact frame when each scene changes, following a strict definition of scene changes as a change of location or time, (2) using these boundaries, the movie video, and the timecoded AD, we label the scene to which each AD segment is describing content. Screenplays for 4/10 of the movies are provided by MovieSum (Saxena & Keller, 2024), and for the remaining 6 we follow Saxena & Keller's (2024) parsing method to obtain parsed screenplays, which involves full manual review and correction. The annotator manually labels the alignment between the movie scenes and the scenes in the screenplays from all ten movies. There was no access to the screenplay to do the scene segmentation, as a result one-to-many and many-to-one alignments are possible. Chapter boundaries are provided directly by our purchased versions of the movies, which we use as sequence boundaries. We obtain professional dialogue subtitles by applying OCR (Tesseract 5.5.0) to the official movie subtitles and manually reviewing (using Subtitle Edit software). We apply this to the English SDH and English dialogue subtitles, or if only the former is available (4/10 cases) then we manually edit the SDH version to remove text other than the dialogue subtitles.

The only potentially subjective task of the above is determining scene boundaries from the movie. We do not perform any double-annotation and hence cannot report inter-annotator agreement on the task. One of the movies in our data is also provided as a part of MovieNet (Huang et al., 2020), which provides data that is the established benchmark for movie scene segmentation. The MovieNet annotation of scene segmentation is crowd-sourced and based on the decisions of an annotator (who likely has not watched the full movies) provided access only to stills from neighbouring shots (determined via automatic shot detection). For a movie in both datasets, *Les Miserables* (2012), we were surprised to find their annotation reports 228 scenes, while ours has only 113. We inspected their boundaries and found them to consistently oversegment, in particular in settings when neighbouring shots are visually different but in fact just a different view of the same scene. As an independent reference, the screenplay of this movie has 133 scenes (some of which were likely removed for the final version).

E New Data Collection Techniques

Working with screenplays brings challenges, both in terms of parsing the PDFs available of them and aligning them to the final movie.

Screenplays are released as 100+ page PDFs, with a reasonably standard scene-segmented structure that is displayed via indentations and font formatting. The task of parsing such documents into their structured contents has been addressed by prior work (Turetsky & Dimitrova, 2004; Agarwal et al., 2014; Winer & Young, 2017). However, the task remains challenging due to different formatting styles (Gil et al., 2011) and in practice several hours of human supervision is needed per movie for perfect parsing (Saxena & Keller, 2024). We do not tackle this task and instead rely on the gold-standard machine-parsed screenplays from the MovieSum dataset (Saxena & Keller, 2024).

However, we found current alignment methods to be insufficient for our purposes. In our approach to screenplay alignment, we define the task at the scene-level as follows: given $\mathcal{S} = \{s_i\}_{i=1}^{S}$ scenes from a machine-parseable screenplay and the dialogue and AD

from $\mathcal{M} = \{m_j\}_{j=1}^{M}$ scenes in the final movie, compute the alignment $A \subseteq \mathcal{S} \times \mathcal{M}$. The screenplay's structured format provides exact segmentation into the S scenes, but scene segmentation is not available for the final movie. Our insight here is that AD will frequently cue movie scene changes; this is a requirement stipulated in both American and British AD guidelines. If we can detect such cues, then we have a high-precision and scalable method of determining movie scene boundaries.

E.1 Movie-Screenplay Alignment

Evaluation and Baselines. Our evaluation metric for movie-screenplay alignment is weighted F_1 score, where the weights are based on the lengths of the scenes in the movie. We will compare our methods against the previous state-of-the-art: aligning a movie's dialogue subtitles with dialogue in the movie's screenplay by computing a dynamic time warping (DTW) over word tokens. In this baseline, we assign any screenplay dialogue word with an exact match to a movie dialogue word the scene the movie word is from.

DTW over word matches is frequently applied for the task of movie-screenplay alignment (Everingham et al., 2006; Sivic et al., 2009; Bouritsas et al., 2018; Papalampidi et al., 2021). Nevertheless, results from Lambert et al. (2013) suggest that DTW may not be the ideal algorithm, because it favours on-diagonal matches in the alignment. This is a particular problem for screenplay drafts before a post-production or even shooting version. Lambert et al. (2013) propose to upweight the score that results from matches; this is akin to minimizing edit distance, which we will employ as a second baseline.[2]

Vecalign-based Alignment. In our approach, we compute embeddings for each scene in the movie and each scene in the screenplay, and then use dynamic programming to align the two. We additionally treat the potential for a screenplay scene to have been removed at production, or for a new scene to be added at production. Thompson & Koehn (2019) provide methods designed for compiling machine translation corpora that can be of use here, which they call Vecalign. In particular they define a cost in the alignment algorithm which allows scenes to be skipped in either the movie or the screenplay, without alignment. We use a fixed skip cost $c_{skip} = 0.4$. Vecalign also treats one-to-many and many-to-one alignments by allowing for merge operations between contiguous sequences of scenes. We use mean-pooling as the merge operation. We allow for alignment blocks of up to 10 scenes. Vecalign also allows for, albeit discourages in the cost function, many-to-many alignments; we disallow these via search constraints so that the final alignment is strictly monotonic over the movie scenes. Vecalign also incorporates methods that reduce the alignment time complexity from quadratic to linear via a recursive approximation; we do not apply these methods and optimize over the full alignment.

The cost $c(s, m)$ of aligning a candidate block of screenplay scenes $s \subseteq \mathcal{S}$ to a candidate block of movie scenes $m \subseteq \mathcal{M}$ is defined as:

$$c(s,m) = \frac{\left(1 - \cos(\bar{\mathbf{s}}, \bar{\mathbf{m}})\right) \cdot |s| \cdot |m|}{\frac{1}{2M}\sum_{j=1}^{M}\left(1 - \cos(\bar{\mathbf{s}}, \mathbf{m}_j)\right) + \frac{1}{2S}\sum_{i=1}^{S}\left(1 - \cos(\mathbf{s}_i, \bar{\mathbf{m}})\right)} \tag{1}$$

where boldface denotes scene embeddings, bar notation denotes embeddings after mean-pooling, and $|\cdot|$ denotes the number of scenes in a block. The numerator scales the similarity cost by the number of scenes in the block alignment, and the denominator normalizes by the mean similarity between the candidate block and all scenes.

With initialization $D(0,0) = 0$, $D(i,0) = i \cdot c_{\text{skip}}$, and $D(0,j) = j \cdot c_{\text{skip}}$, the dynamic programming recurrence relation is as follows:

$$D(i,j) = \min \begin{cases} D(i-1,j) + c_{\text{skip}} & \text{(skip screenplay scene)} \\ D(i,j-1) + c_{\text{skip}} & \text{(skip movie scene)} \\ \min\limits_{|s|,|m|} \left[D(i-|s|, j-|m|) + c\left(\mathcal{S}_{i-|s|:i}, \mathcal{M}_{j-|m|:j}\right)\right] & \text{(block match)} \end{cases}$$

[2]We use the implementation from https://github.com/jitsi/jiwer

Method	Data	Model	P	R	F_1
Text-based DTW	dialogue	—	63.7	62.4	62.3
Text-based MED	dialogue	—	66.4	63.1	63.8
Vecalign	dialogue	Qwen3 E 8B	81.4	53.4	64.2
Vecalign	SDH	Qwen3 E 8B	82.8	58.5	68.0
Vecalign	UK AD	Qwen3 E 8B	76.9	67.1	71.3
Vecalign	US AD	Qwen3 E 8B	77.0	70.0	73.0
Vecalign	UK AD + US AD	Qwen3 E 8B	76.9	72.8	74.5
Vecalign	dialogue + UK AD	Qwen3 E 8B	82.1	73.7	77.3
Vecalign	SDH + UK AD	Qwen3 E 8B	83.6	73.1	77.5
Vecalign	dialogue + US AD	Qwen3 E 8B	84.7	73.9	78.6
Vecalign	SDH + US AD	Qwen3 E 8B	85.1	73.9	78.7
Vecalign	dialogue + UK AD + US AD	Qwen3 E 8B	83.1	77.4	79.9
Vecalign	SDH + UK AD + US AD	Qwen3 E 0.6B	87.5	62.4	71.8
Vecalign	SDH + UK AD + US AD	Qwen3 E 4B	85.3	73.8	78.8
Vecalign	SDH + UK AD + US AD	Qwen3 E 8B	84.9	77.7	**80.9**

Table 6: Screenplay alignment results. DTW = dynamic time warping, MED = minimum edit distance. Text-based baselines are in first block, ablation results are in second block, and best results based on embedding models of three sizes are in final block.

where the search space is constrained such that $\min(|s|, |m|) = 1$ and $|s| + |m| \leq 10$. The final alignment A is then recovered by backtracking along the minimal-cost path from $D(S, M)$ to $D(0, 0)$ and collecting the matched scenes.

Embeddings. For the movies, we will define the embeddings based on a time-sorted concatenation of the subtitles (SDH or dialogue-only) and/or AD (UK and/or US versions) in each scene in our gold-standard annotation; for the screenplay we define the embedding based on the dialogue and scene descriptions in each scene in the order that they appear. Embeddings are computed according to Zhang et al. (2025).

Results. Results are in Table 6. The word-based DTW baseline achieves $F_1 = 62.3$, and the minimum edit distance variant achieves $F_1 = 63.8$. Performance with Vecalign applied to the same dialogue is comparable at $F_1 = 64.2$. Using instead SDH leads to an improved $F_1 = 68.0$ (+3.8), demonstrating that the non-verbal audio captions contain useful information for the alignment. We can outperform these dialogue-based alignment approaches using only one of UK or US AD ($F_1 = 71.3$ and $F_1 = 73.0$), demonstrating that AD also contains useful information for the alignment. Using both AD versions results in better performance than using only one ($F_1 = 74.5$, +3.2 and +1.5 for AD versions). (Table 6 also gives results for other combinations of input data.) Our best results (F_1=80.9) are based on using SDH and both AD versions with Vecalign. In our experiments so far, we have used the Qwen3 Embedding 8B model for computing the embeddings. Using the 4B or 0.6B variants of this model leads to worse performance (F_1=78.8 and F_1=71.8, respectively). We apply this best alignment pipeline to all movies with screenplays in **RE FRAME D**.

E.2 Movie Scene Segmentation from AD

AD scene segmentation. We detect scene change cues in AD by finetuning RoBERTa-large (Liu et al., 2019) on our manually annotated data. The task is for a model to classify each segment of AD as starting a new scene or not, given the context of surrounding AD segments. Our evaluation metric is binary F_1 score. We perform 10-fold cross-validation and report the macro-average. Our baseline will assume that all AD segments that begin with the word 'Now' (this is a word US AD guidelines specify can be used to cue scene changes) start a new scene.

Parameter	Value
Block size	512
Learning rate	1e-5
Batch size	64
Epochs	3
Weight decay	0.01
Warmup ratio	0.06

Table 7: Hyperparameters for the scene segmentation fine-tuning.

Parameter	Value
Block size	128
Learning rate	1e-5
Batch size	32
Epochs	3
Weight decay	0.01
Warmup ratio	0.06

Table 8: Hyperparameters for the token classification AD splitting model.

		P	R	F_1
Baseline		—	$11.9_{\pm 0.0}$	—
RoBERTa (base)	sin.	$59.3_{\pm 0.4}$	$55.3_{\pm 0.1}$	$56.6_{\pm 0.2}$
	dou.	$73.2_{\pm 1.1}$	$45.4_{\pm 0.3}$	$55.5_{\pm 0.4}$
RoBERTa (large)	sin.	$65.0_{\pm 0.2}$	$\mathbf{55.8}_{\pm 0.1}$	$\mathbf{59.5}_{\pm 0.2}$
	dou.	$\mathbf{76.1}_{\pm 0.4}$	$49.2_{\pm 0.7}$	$\mathbf{59.2}_{\pm 0.4}$

Table 9: Scene segmentation performance (binary F_1, precision P, recall R; 10-fold CV, $\pm$ std). *sin.*: per-version training; *dou.*: joint US/UK training for congruent boundaries. Baseline: segments starting with "Now" begin a new scene (P undefined as some tracks never use it).

	P	R	F_1
Baseline	$38.0_{\pm 0.0}$	$38.4_{\pm 0.0}$	$36.9_{\pm 0.0}$
RoBERTa (base)	$64.9_{\pm 0.1}$	$66.2_{\pm 0.0}$	$63.7_{\pm 0.1}$
RoBERTa (large)	$\mathbf{65.8}_{\pm 0.0}$	$\mathbf{68.5}_{\pm 0.1}$	$\mathbf{64.8}_{\pm 0.0}$

Table 10: AD splitting performance (character-level F_1, precision P, recall R; 10-fold CV, $\pm$ std). Task: classify whether each character ends a description element. Baseline: split at every comma.

We compare two formulations of the training task. The first is called 'single' and involves performing training and inference on independent sliding windows of six consecutive segments from individual AD transcripts. The second is called 'double' and involves performing training and inference on paired sliding windows from US and UK transcripts to predict congruent boundaries across both. For 'single', we collect many samples of '[CLS] 3 segments [SEP] 3 segments' based on a sliding window of 6 segments. The learning task is binary classification on the CLS token of whether the [SEP] token represents a scene boundary or not. We collect all possible samples independently of each of the US and UK AD transcripts. 'Double' utilizes our prior that both the US and UK AD versions are describing the same movie scene changes. We formulate the task so that the model can only predict congruent scene boundaries in both AD versions. The approach involves jointly training on sequences of US and UK AD using the following template: '[CLS] 3 segments of US AD [SEP] 3 segments of UK AD [SEP] 3 segments of US AD [SEP] 3 segments of UK AD'. The two versions may not begin narrating a scene at the same time. In fact, in 8.7% of scene boundaries in our data, after the first AD version starts a new scene, the other AD version starts at least one new segment of the previous scene, before moving to the next scene. The boundary is fuzzy, and with this approach and two sliding windows, we can treat this. We allow for the sliding windows to separate at the start of segment number four (the potentially new scene) up to 10 seconds, or the next AD if there is none within 10 seconds for one of the versions. The training task is now whether the second and third [SEP] tokens represent scene boundaries to the same next scene. To ensure the predicted scene boundaries are consistent, we use greedy inference, discarding predictions that are incompatible with earlier ones. We do not want our model to learn any signals present in our gold-standard AD transcripts, but not in automatic ASR transcripts. Hence we train on transcripts from Speechmatics, segmented according to the gold-standard segmentation. The full set of training hyperparameters is provided in Table 7.

Results are in Table 9. The baseline approach achieves a recall score of 11.9, but precision is undefined since some of the movie ADs never use 'now'. The 'single' and 'double' approaches with the RoBERTa-large model achieve similar F_1 measures of 59.5 and 59.2,

respectively. Using the smaller RoBERTa-base model results in 56.6 and 55.5. Precision and recall metrics show that they are behaving differently: 'double' achieves higher precision, yet lower recall. This is intuitive, since the 'double' relies on a model's prediction that a scene change is distinguishable in both transcripts. Since 'double' brings the additional benefit of the scene boundaries guaranteed to be congruent in the two versions, we consider it best and use it. For the earlier video extract from *The Girl with the Dragon Tattoo*, this model successfully segments into its three constituent scenes: at a tailor's shop where the gift is purchased, at Salander's apartment where she writes a card, and the final street scene in Figure 1; as a result, the Vecalign-based alignment is able to successfully align each of these three scenes with its corresponding scene in the screenplay.

Incorporating video scene boundaries. The above procedure provides congruent scene labels for each AD segment in our two AD versions. **RE FRAME D** contains AD for individual video scenes, which can also provide a signal of scene boundaries. In order to assign AD and subtitle segments to each video, we include all segments that are fully contained in the temporal span of the video. Furthermore, for the screenplay alignment to assign a particular video a number of screenplay scenes, we must enforce that scenes do not span over the boundary of the start or end of a video – in such cases, we need to divide the scene labels further. We also wish to transfer these scene labels to the available dialogue and SDH data. Since some videos in the training set have temporal overlap, we need a general re-assignment algorithm that caters to this.

Our scene re-assignment algorithm meets all of the following requirements: (1) scene boundaries according to the AD classifier are retained; (2) temporal scene boundaries are created for the start and end of every video; and (3) the data associated with each video can be reconstructed by concatenating the data associated with a particular set of scene labels.

This is achieved by constructing an intermediate scene label for every segment before mapping these labels to final shared scene identifiers. Each AD segment is initialized with the scene label assigned by the AD classifier. We then append the identifiers of all videos in which the segment is fully contained. To distinguish different positions relative to video boundaries, we partition the movie timeline at every video start and end time. Each resulting interval is associated with the set of videos active during that interval, which may be none. For each segment, we assign the interval whose active video set matches the segment's fully contained video set, when such an interval is unique. If no unique matching interval exists, the segment is assigned a boundary-spanning interval covering the relevant video-boundary regions. Thus, each AD segment receives an intermediate label consisting of its classifier scene, its fully contained video set, and its position relative to the global video-boundary partition.

For subtitle and SDH segments fully contained in videos, we infer the classifier scene from AD segments in the same video. When multiple AD scenes occur in the video, the subtitle is assigned to the one with greatest temporal overlap. For subtitle and SDH segments outside videos, we infer the classifier scene from global intervals derived from the first occurrence of each AD classifier scene across both AD versions, partitioning the full movie timeline from zero to the latest segment or video end. The classification again uses greatest temporal overlap. If a video contains subtitle or SDH segments but no AD segments, we add a subtitle-only scene label for that video setting. Finally, all intermediate labels across the two AD versions, subtitles, and SDH subtitles are mapped to shared consecutive scene identifiers in order of their first occurrence. For the screenplay alignment, each movie scene of length less than 10 seconds and not a part of any video is excluded from the alignment.

E.3 AD Splitting

ASR systems output transcripts at the sentence level, which are potentially long (the longest in our data is 74 words). To be consistent with our evaluation data, we would like to segment transcribed AD sentences into the separate elements they describe. This task is only defined fully with access to the movie visuals, but we expect that reasonable performance is possible using only the textual sentences as input. Our operationalisation draws inspiration from Frohmann et al. (2024), who formulate the task of text segmentation as classifying whether

each token in a text ends a segment or not. We create training data by splitting Speechmatics AD transcript sentences according to the segmentation in our gold standard. This results in 16,496 sentences split into 21,675 segments (26.45% of sentences have more than one segment). We evaluate using character-level F_1 on segment-ending characters. Our baseline creates an AD split at every comma. Table 8 provides the full set of hyperparameters used.

Results are in Table 10. The comma baseline achieves F_1=36.9, while our best model achieves F_1=64.8. Evidently the task is not defined perfectly, likely in part because of our text-only formulation, but we deem this satisfactory performance. Manual inspection shows that the resulting model is quite robust to punctuation and complex syntax. For example, the AD 'His right hand forms a cannon, which he aims down the corridor before inserting a probe into the mainframe terminal at a radio telescope array, three dishes shift their positions.' becomes 4 segments (starting 'His', 'which', 'before' and 'at'). This example also demonstrates how preprocessing our data with this segmentation model is an important pre-processing step before running the scene segmentation model, because the segment beginning 'at' actually starts a new scene.

F More Details of AD Transcription Experiments

We evaluate ASR-only performance using gold-standard diarization; metrics include segment error rate (SER) and word error rate (WER) (which is derived from substitution rate (SR), deletion rate (DR) and insertion rate (IR)). Binary diarization error rate (DER) gives diarization performance; it is derived from miss rate (MR) and false alarm rate (FAR). We evaluate the error rate of timestamps against collars of α seconds around the reference timestamps. Character names in the reference are manually identified and character name error rate is a micro average over reference characters spelt correctly. Finally, overall ASR metrics show end-to-end performance, i.e. after automatic diarization. All text is pre-processed according to Radford et al. (2023).[3]

We can also evaluate the quality of the MAD-v2-eval transcripts against our new reference. The models we compare are as follows:

- **The system used for AD transcription in MAD-v2-train and CMD-AD datasets** (Han et al., 2023a; 2024)**: WhisperX.** It is an open-source system that combines multiple open-source projects. We also tried customizing the system to use `gpt-4o-transcribe` for the ASR.
- **The system used for AD transcription in the MAD-v1-train dataset** (Soldan et al., 2022)**: Microsoft Batch API system.** Since our data consists of both American English and British English accents. We investigated variants with and without continuous language identification between British and American English; in summary, including this language identification brings substantial performance gains. We show results including using this language identification.
- **A system previously not applied to the task, but generally-reported good performance: Speechmatics Batch API** `standard` and `enhanced` systems, global English.

Characters play a central role in movie narratives, and hence it is vital that they are transcribed correctly. We apply pre-existing methods for guiding ASR models to spell particular words correctly, which for us is character names. We collect lists of all character names in movie credits, except those that are non-unique (e.g. 'Citizen') or ending in a numeric character (e.g. 'Citizen 1').[4]

- Whisper: we override Whisper's in-context tokens with a script containing all character names. We compared three different script templates and proceeded with the best one: a comma separated list of the character names, which outperformed specifying the name of the movie and a template more realistic of a real conversation.
- Microsoft: vocabulary adaptation is not available in Batch API.

[3] https://github.com/openai/whisper/tree/main/whisper/normalizers

[4] These are available on 'fullcredits' pages on IMDb.

		ASR					Diarization			Names		Times			Overall			
		SER	WER	SR	DR	IR	DER	MR	FAR	Micro	Macro	0.2	0.4	0.5	WER	SR	DR	IR
WhisperX	`large-v3-turbo`	23.5	6.6	2.9	2.7	1.0	14.9	3.7	11.3	22.3	22.2	5.1	1.0	0.8	13.7	3.6	2.4	7.7
	`gpt-4o-transcribe`	49.0	38.6	3.2	34.6	0.7	42.0	34.8	7.3	51.4	54.9	9.6	5.3	5.1	37.9	4.6	32.0	1.3
	large-v3	23.5	8.0	3.1	3.8	1.1	16.7	4.7	11.9	21.7	21.7	5.2	1.1	0.9	14.6	4.2	3.1	7.3
	large-v2	27.3	11.3	4.0	6.0	1.4	17.5	6.8	10.7	28.4	27.4	5.3	1.2	1.0	17.0	5.6	4.7	6.7
	large	23.5	8.0	3.1	3.8	1.1	16.7	4.7	11.9	21.7	21.7	5.2	1.1	0.9	14.6	4.2	3.1	7.3
	medium	27.4	10.5	4.2	5.1	1.3	16.7	5.9	10.8	28.1	29.2	5.2	1.1	0.9	16.3	5.8	3.9	6.7
	medium.en	25.1	7.2	3.1	2.9	1.1	14.4	4.0	10.4	23.9	23.7	5.3	1.1	1.0	14.1	3.7	2.6	7.7
	small.en	28.1	7.6	3.8	2.7	1.2	14.3	3.3	11.0	25.3	25.0	5.1	1.0	0.8	14.6	4.3	2.5	7.8
	small	29.6	8.4	4.4	2.6	1.4	13.9	3.3	10.6	27.4	28.4	5.3	1.2	1.0	15.3	5.0	2.3	7.9
	base.en	35.1	10.1	6.1	2.2	1.8	13.8	2.7	11.1	30.7	31.8	5.0	0.9	0.8	17.2	6.6	2.1	8.6
	base	40.0	12.9	7.5	3.2	2.2	14.1	3.4	10.7	32.9	35.7	5.1	1.1	0.9	19.6	8.2	2.9	8.6
	tiny.en	43.7	14.4	8.9	2.9	2.6	13.4	3.1	10.3	36.4	38.6	5.0	0.9	0.8	21.2	9.6	2.5	9.2
	tiny	51.5	19.1	11.7	3.6	3.8	14.4	3.8	10.7	41.1	45.0	5.1	1.1	1.0	25.6	12.4	3.1	10.2
Microsoft	`standard`	32.5	7.5	4.5	1.2	1.8	20.6	0.4	20.2	25.9	26.1	2.8	0.3	0.2	17.8	4.6	0.6	12.6
Speechmatics	`standard`	25.8	5.6	3.7	0.9	0.9	3.0	0.5	2.5	22.9	24.4	1.9	**0.0**	**0.0**	5.8	3.6	0.9	1.3
	`enhanced`	**18.3**	**3.6**	**2.3**	**0.7**	**0.6**	**2.5**	**0.3**	**2.2**	**16.2**	**17.8**	**0.7**	**0.0**	**0.0**	**3.6**	**2.3**	**0.5**	**0.8**

Table 11: AD transcription results *without* character-name adaptation (%, lower better; best in **bold**). Columns as in Table 4, with additionally a macro-averaged character-name error rate and a 0.5s timestamp collar. Full per-system version of Table 4; adapted counterpart in Table 12.

		ASR					Diarization			Names		Times			Overall			
		SER	WER	SR	DR	IR	DER	MR	FAR	Micro	Macro	0.2	0.4	0.5	WER	SR	DR	IR
WhisperX	`large-v3-turbo`	17.6	5.4	1.7	2.6	1.1	14.1	3.4	10.7	7.2	9.6	5.3	1.0	0.9	12.8	2.2	2.5	8.1
	`gpt-4o-transcribe`	34.1	22.3	2.5	18.8	1.0	30.7	19.0	11.7	30.7	32.6	8.0	3.8	3.5	24.4	4.3	16.5	3.6
	large-v3	18.4	8.1	1.9	5.2	1.0	16.1	5.9	10.2	8.7	11.0	5.7	1.5	1.3	14.2	2.7	4.6	6.8
	large-v2	20.2	10.2	2.4	6.7	1.1	18.1	7.4	10.7	12.0	14.9	5.7	1.7	1.4	15.4	3.8	5.5	6.2
	large	18.4	8.1	1.9	5.2	1.0	16.1	5.9	10.2	8.7	11.0	5.7	1.5	1.3	14.2	2.7	4.6	6.8
	medium	22.4	11.3	2.6	7.6	1.1	18.4	8.2	10.3	12.1	14.1	5.8	1.6	1.4	15.8	4.1	6.2	5.5
	medium.en	19.5	8.1	1.7	5.4	1.0	15.9	6.2	9.7	8.3	11.3	5.7	1.6	1.4	13.6	2.4	4.8	6.3
	small.en	21.5	7.3	2.4	3.8	1.1	14.7	4.2	10.5	7.4	10.8	5.5	1.3	1.2	13.4	3.0	3.5	7.0
	small	23.5	9.0	2.8	5.0	1.2	16.0	5.7	10.3	8.8	14.5	5.6	1.4	1.2	15.1	3.6	4.5	7.0
	base.en	28.5	8.5	4.4	2.6	1.4	13.6	3.0	10.6	11.5	13.2	5.3	0.9	0.8	15.3	5.1	2.4	7.7
	base	32.0	10.0	5.5	2.6	1.8	13.6	3.0	10.6	8.9	14.8	5.1	0.9	0.8	16.9	6.3	2.3	8.4
	tiny.en	36.8	12.3	7.2	2.8	2.4	13.8	3.0	10.8	12.4	15.0	5.2	1.0	0.8	19.1	7.9	2.4	8.8
	tiny	46.0	17.8	9.7	4.9	3.2	15.9	5.1	10.8	19.1	21.6	5.5	1.3	1.2	23.6	10.9	3.9	8.8
Speechmatics	`standard`	23.2	4.9	3.1	0.9	0.9	3.1	0.4	2.7	15.6	14.5	1.9	0.0	0.0	5.2	3.1	0.8	1.3
	`enhanced`	**14.0**	**2.8**	**1.5**	**0.7**	**0.6**	**2.7**	**0.2**	**2.5**	**4.8**	**7.9**	**0.8**	**0.1**	**0.0**	**2.9**	**1.5**	**0.5**	**0.8**

Table 12: AD transcription results *with* character-name adaptation (%, lower better; best in **bold**). Columns as in Table 11. Character-name-adapted counterpart of Table 11; adaptation details in Appendix F.

- Speechmatics: API-based vocabulary adaptation with the full list of character names, including both complete names and their constituent parts.

For a full set of numerical results, see Table 11 for without character name adaptation, and Table 12 with character name adaptation.

We performed a spot check of the remaining mistakes in the Speechmatics transcription. The majority of the mistakes fall into one of four categories: (1) remaining character misspellings (particularly for non-standard name spellings), (2) incorrectly transcribed articles (sometimes they are dropped entirely, sometimes swapped for other articles), (3) incorrectly transcribed homophones, and (4) imperfections in the evaluation normalization script (particularly with relation to compound words). Furthermore, we found the output to often be sentence-segmented incorrectly. We provide a remedy for this for **REFRAMED** based on applying our AD splitting model (see Section E.3).

G Subtitles

Our approach matches movie dialogue ASR with subtitle files available online, and allows for a constant shift between the two. We collect English SDH and dialogue-only subtitles from `opensubtitles.org`. The data provides many subtitle versions for each movie — we

assign the highest quality score to subtitles with labels 'trusted' or 'subtranslator' and otherwise scores based on the subtitle's download count. We use the non-AD transcription from our AD track transcripts. For each movie, we try candidate subtitles in decreasing order of quality score, accepting the first where at least 90% of matched words have timestamps within 1 second of the ASR transcript, requiring a minimum of 50 matched timestamp pairs.[5] For each candidate subtitle we try first without applying a constant offset, and if that fails the two conditions, we add the offset that maximises the proportion of inliers.

We apply a data cleaning pipeline to the resulting subtitles, as well as manual review (see also Section H).

- We apply the 'fix common errors' tool available in Subtitle Edit v5.0.0, disabling only steps that would alter line breaking or sentence casing.
- Next we normalize the text by removing text that concerns renderer state, collapsing repeated spacing and adding spacing after leading hyphens (speaker-change format in subtitles). We correct a recurring error in which the capital letter I has been transcribed as a lowercase l within otherwise fully uppercase segments.
- We manually review segments with URLs, and other potential attribute identifiers and remove if it is deemed that these are not part of the original subtitles.
- We enforce a minimum inter-segment gap of 24 ms and require strictly positive duration on every segment. Offending segments are first trimmed against the window defined by their neighbours; if trimming yields an empty interval, the segment is re-timed at a target reading rate of 20 characters per second, borrowing up to 200 ms from an adjacent segment longer than 500 ms as a last resort.

H Dataset Compilation Review Steps

For the scenes in our dataset, we have reviewed all titles manually and filtered out videos which are not scenes from a final movie (e.g. deleted scenes, trailers, interviews). We manually determine which speaker label corresponds to the AD narrator. Based on a manual review of the transcripts, we keep movies with two AD transcripts that appear to be created independently (as opposed to being identical or one being an edit of the other). Based on manual review of the dialogue and SDH subtitles, we found 3 marked as normal which were in fact SDH and 4 subtitles marked as SDH which were in fact normal; only the remaining are kept. The US AD tracks are all at the same speed as the videos; the UK AD tracks either require a PAL-NTSC speed-down, or are at the same speed. We keep movies for which $\geq$80% and $\geq$ 5 of the videos have the same predicted speed difference, and only keep videos with this predicted speed difference. We manually remove AD transcription after the start of movie end credits. Before sending the validation and test split movie scenes with their automatically aligned AD audio tracks to the company performing manual transcription, we reviewed the videos for their alignment. Out of 106 scenes in the test set, two had imperfect alignment and were discarded; out of 99 in the validation set, three were discarded. For these evaluation scenes, we also remove those with any kind of advertisement (typically a couple seconds at the end promoting the physical release); 4/106 in the test set and none in the validation set. Finally, for these scenes we manually reviewed scenes from the same movie that had any temporal overlap and discard scenes until this is not the case; 9/106 from the test set and none from the validation set. The final test set has 91 scenes and the final validation set has 96 scenes.

I Dataset Splits

Genre labels are obtained from IMDb, where each film is assigned one or more genre descriptors. Our partitioning into train, validation and test splits employs a distributional stratification strategy to align the label frequencies of the validation and test sets with the

[5]This is inspired by the popular (but now unfortunately deprecated) tool 'subsync', as used for the MovieNet dataset (Huang et al., 2020)).

training set. For each evaluation set, 10 movies are selected from the pool of candidate movies by minimizing the Jensen-Shannon divergence between the samples and the train genre distribution. Acceptable solutions may have zero representation of a genre that is represented in the train set, hence our choice of Jensen-Shannon divergence. Samples not selected for test are returned to the train set; we do not do this for validation, as these were movies used in prior datasets (LSMDC, MAD and CMD-AD). This is formalized in Algorithm 1 and the final genre split is visualized in Figure 7.

Algorithm 1 Genre stratification via Jensen-Shannon divergence

Input: Initial partitions $D_{\text{train}}, D_{\text{pool_val}}, D_{\text{pool_test}}$; desired subset size k
Output: Final splits $S_{\text{train}}, S_{\text{val}}, S_{\text{test}}$
$\mathcal{L} \leftarrow$ unique labels in D_{train} ▷ Target label vocabulary
$P \leftarrow \text{Normalize} \sum_{i \in D_{\text{train}}} \text{OneHot}_{G_i,\mathcal{L}}$ ▷ Target genre distribution
function FINDBEST(D_{pool}, P, k)
  $\delta_{\min} \leftarrow \infty$
  for subset $s \in \text{Combinations}_{D_{\text{pool}},k}$ **do** ▷ Iterate through all possible subsets
    $Q \leftarrow \text{Normalize} \sum_{j \in s} \text{OneHot}_{G_j,\mathcal{L}}$ ▷ Candidate subset distribution
    $M \leftarrow 0.5P + 0.5Q$ ▷ Mixture distribution
    $\text{kl}_{\text{target}} \leftarrow 0.5 \sum P_i \log \frac{P_i}{M_i}$ ▷ Target divergence from mixture distribution
    $\text{kl}_{\text{subset}} \leftarrow 0.5 \sum Q_i \log \frac{Q_i}{M_i}$ ▷ Subset divergence from mixture distribution
    $\text{js} \leftarrow \text{kl}_{\text{target}} + \text{kl}_{\text{subset}}$ ▷ Total symmetric Jensen-Shannon distance
    if $\text{js} < \delta_{\min}$ **then**
      $\delta_{\min} \leftarrow \text{js}, s^* \leftarrow s$ ▷ Keep subset that best approximates target
    end if
  end for
  return s^*
end function
$S_{\text{val}} \leftarrow \text{FindBest}_{D_{\text{pool_val}},P,k}$
$S_{\text{test}} \leftarrow \text{FindBest}_{D_{\text{pool_test}},P,k}$
$S_{\text{train}} \leftarrow D_{\text{train}} \cup D_{\text{pool_test}} \setminus S_{\text{test}}$ ▷

J Face Detection Labeling

Every character mention in both of the two references of the test split scenes was manually labeled without use of automatic tools. If a character is named differently in the two references, we selected one.[6] We then run an off-the-shelf face-detection tool[7] on frames sampled at one frame per second on the input video. The first discernable face corresponding to each character in each video was annotated. Of 315 character-video pairs, 268 were successfully labeled. The remaining concern cases when the character's face is never seen directly in the video (e.g. only from behind), the character is seen only for less than 1 second at a time in the video, or the face detection tool failed to recognize the face. Of these unlabeled cases, 68.1% have a labeled face from another video from the same movie which future work could use. We maintain our closed-loop design and do not prompt with the names of unlabeled characters.

[6] There were six cases of this (three first name vs. surname, three acronyms/nicknames).
[7] https://github.com/ageitgey/face_recognition

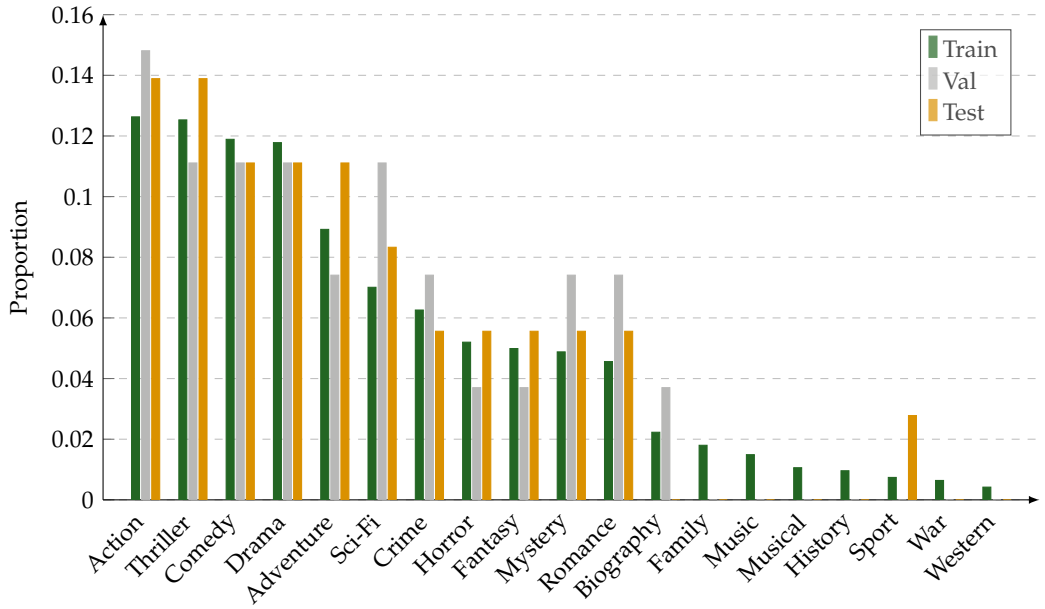


Figure 7: The final distribution over genres in each of train, validation and test splits in **REFRAMED**. We have applied an approach that maximises the distributional similarity over genres across the three splits.

K Example US AD, UK AD and Screenplay Scene

US AD

a. Night-time.
b. Snow drifts to the ground
c. as Salander glides her motorcycle down a hill
d. and parks across the street
e. from Blomkvist's apartment building.
f. Dismounting, she removes her helmet,
g. then takes the garment bag with her gift
h. off the bike's rear cargo rack.
i. She sees Blomkvist and Erika leaving the building
j. and watches unnoticed as he drapes
k. an arm over her shoulders.
l. The couple heads down a walkway
m. to a waiting taxi at the top of a hill.
n. Salander watches woundedly
o. as they climb into the cab,
p. then lowers her heartbroken gaze.
q. Turning away, she tosses Blomkvist's gift
r. into a dumpster against the side of a building.
s. She glances at the cab as it drives off,
t. then dons her helmet
u. and mounts her bike.
v. She speeds off down the cobblestone street,
w. her figure disappearing from view as she rounds a corner.

UK AD

a. Night.
b. Lisbeth rides her motorcycle down a cobbled street
c. with snow piled at the curbside.
d. She parks on a corner outside Mikael's apartment
e. and removes her crash helmet.
f. She takes a package from the back of the bike,
g. but stops dead when she sees Mikael
h. leave his apartment with Erika.
i. She watches numbly as the couple walk away
j. with their arms around each other.
k. Lisbeth looks down as Mikael and Erika
l. get into a waiting taxi.
m. Lisbeth turns away and flings the package which has the card
n. attached to it into a nearby dumpster.
o. She puts her crash helmet back on,
p. gets on her bike
q. and starts it up.
r. A solitary figure, she rides off into the night.

```
<scene>
<stage_direction>EXT. STOCKHOLM - EVENING</stage_direction>
<scene_description>Snow's falling, but unlike the brutal winter storms last year,
    it's just heavy enough to dust the city in white powder that reflects the
```

```
    Christmas lights in the trees. The beauty of her city pleases Salander as she
    rides through it on her motorcycle. Or maybe it's something she's feeling as
    she turns onto Blomkvist's street. She parks. Gathers the garment bag and card,
    walks under construction scaffolding toward his apartment building. From around
    the corner ahead, Blomkvist and Erika appear. He says something and she laughs,
    puts her arm around his waist and lays her head against his neck scarf.
    Salander stops mid-stride, ducks into an alcove under the scaffolding, waits as
    long as it should take them to get inside the building, then peers out again -
    They're not inside. They've stopped just outside the building's entrance.
    They're embracing. And the longer they stand there together, the sharper the
    pain stabs at Salander. Finally they separate, but only to allow Blomkvist to
    unlock the door. Erika takes his hand then, and they disappear inside. Salander
    can't move. Waits for the pain to subside - which it eventually does - but only
    to be replaced with a sense of helplessness. Then that subsides, replaced with
    a facade of insouciance. The richest girl in Sweden emerges from the alcove and
    walks back the way she came - tossing the garment bag and card into a
    construction dumpster - climbs onto her Honda - starts it - and drives off -
    Probably forever.</scene_description>
</scene>
```

L Further Details on Evaluation Metrics

For the test set, evaluation is performed at the level of each video; for the challenge set, at the sequence-level. Movie-level scores are the mean of the video or sequence scores, and the final metrics are a macro-average across all movies. References are already segmented into description elements; generations should also be defined to be at the level of individual visual description elements.

As evaluation preprocessing, we segment all generations into sentences with the `sat-12l-sm` model from Frohmann et al. (2024). We then further segment each resulting sentence into description elements using our learned AD splitting model. We re-purpose the splitting modeling used for the training set of **RE FRAME D**: re-training directly on the gold AD boundaries (as opposed to boundaries mapped onto the Speechmatics transcripts). To enhance the robustness of this splitting model, we apply the model initialization and data augmentation that Frohmann et al. (2024) used for `sat-12l-sm` training. Cross-validation on the gold data shows that this trained model performs similarly to the learned splitting model used for the **RE FRAME D** dataset (F_1=66.0, vs. a comma-splitting baseline F_1=38.4).

L.1 Alignment-based Evaluation

SODA computes an optimal monotonic alignment between generated and reference descriptions, using METEOR as the pairwise content score. The alignment preserves temporal order and allows unmatched generated or reference descriptions. Sorted by start time, let us denote the sequence of n reference description elements as $\mathcal{D} = (d_i)_{i=1}^{n}$ and the sequence of $\hat{n}$ generated description elements as $\hat{\mathcal{D}} = (\hat{d}_j)_{j=1}^{\hat{n}}$. Using the METEOR evaluation metric, we then define the following:

$$M(i,j) = \text{METEOR}(d_i, \hat{d}_j).$$

Let $S(i,j)$ be the best alignment score between the first i reference and first j generated descriptions, computed recursively as:

$$S(i,j) = \max \begin{cases} S(i-1,j) \\ S(i,j-1) \\ S(i-1,j-1) + M(i,j) \end{cases}$$

with $S(i,0) = S(0,j) = 0$ and ties broken in favor of the final case.

Backtracking yields the optimal alignment $\mathcal{A}^\star \subseteq [|\mathcal{D}|] \times [|\hat{\mathcal{D}}|]$. We then compute SODA-M as the F1 score over aligned METEOR content:

$$P = \frac{\sum_{(i,j)\in\mathcal{A}^\star} \text{METEOR}(d_i, \hat{d}_j)}{|\hat{\mathcal{D}}|}, \quad R = \frac{\sum_{(i,j)\in\mathcal{A}^\star} \text{METEOR}(d_i, \hat{d}_j)}{|\mathcal{D}|},$$

$$\text{SODA-M} = \frac{2PR}{P+R} = \frac{\sum_{(i,j)\in\mathcal{A}^\star} \text{METEOR}(d_i, \hat{d}_j)}{\frac{1}{2}\left(|\mathcal{D}| + |\hat{\mathcal{D}}|\right)}$$

For temporal grounding, SODA-T measures whether aligned descriptions are placed close in time. Let $m(I_i)$ denote the midpoint of the temporal interval I_i of description element d_i, and τ the temporal tolerance:

$$\text{SODA-T} = \frac{1}{|\mathcal{D}|} \sum_{(i,j)\in\mathcal{A}^\star} \mathbf{1}\left[|m(I_i) - m(\hat{I}_j)| < \tau\right].$$

Following prior SODA implementations, we take the maximum of each score across references.

L.2 QA Evaluation

We prompt an LLM with each segment of AD and 1 minute transcript of contextual AD (segments that end after 30 seconds before the target segment and end before 30 seconds after the target segment), and instruct the model to generate up to 3 questions and multiple choice answers of what is described. The prompt we used is as follows:

Question Generation Prompt

```
You are provided a transcript of one or more English audio description sentences
from a 1-minute section of a movie. Based on one segment of text contained within
the transcript, your task is to create 0, 1, 2, or 3 English questions and
multiple-choice answers.

These questions will later be used for an evaluation task where the respondent has
access to several minutes of a different audio description transcript of the same
section of the same movie. The respondent will not be told which segment your
question refers to.

Guidelines that your questions must follow:

    1. The question must include sufficient detail to uniquely identify the
    specific aspect under question from the wider context. If the aspect under
    question is strongly related to something also described elsewhere, or
    something that plausibly could happen nearby in the movie, do not create a
    question about it.

    2. The question must ask directly about the movie world (never use
    meta-phrasing such as 'described in', 'segment', 'scene', 'audio
    description').

    3. The question must concern content addressed in the provided segment of
    interest. You may use the full transcript for context, and it is acceptable
    for your question to only be answerable given context.

    4. For "who" questions about characters, always use the pronouns
    they/them/their. Otherwise, if referring to one or more characters, the
    question must always use character names. If a pronoun cannot be resolved to a
    character's name given the transcript context, do not create a question about
    it.
```

```
    5. The question must be phrased in such a way that there can feasibly be 5
    possible answers, only one of which will be correct.

Guidelines that your correct_answer must follow:

    1. The answer should use wording similar to the segment of interest. If
    needed, you may apply simple reformulation so that the correct_answer is
    consistent with all the wrong_answers.

    2. The answer should have a length of between 1-4 words. The number of words
    in an answer is equal to the number of spaces plus one.

    3. If referring to one or more characters, the answer must always use the
    character's names. If a pronoun cannot be resolved to a character's name given
    the transcript context, do not create a question about it.

Guidelines that your wrong_answers must follow:

    1. There must be exactly four wrong answers.

    2. Wrong answers should be between 1-4 words and of similar style as the
    correct answer. If the correct_answer uses a particular article, every wrong
    answer should use it too. The correct answer should not stand out because of a
    surface feature.

    3. Wrong answers should all be both plausible and absolutely wrong. You should
    create them using your general knowledge. The correct answer should not stand
    out because by guessing it is more plausible than your wrong_answers.

Before your final response, check that all of the above conditions are met. If
there are no suitable questions, it is fully acceptable to respond with zero
questions. If only 1 or 2 questions are possible, only respond with this many
questions. There is no penalty for creating fewer than 3 questions.

Your response must be a valid JSON list containing up to three objects. Each of
these objects must have three keys: "question" mapped to a string of your
question, "correct_answer" mapped to a string of your correct answer, and
"wrong_answers" mapped to an array of exactly four strings of your wrong answers.

Example

Full transcript: Bright light fills our view and the caretaker's tea kettle
steams. In a bedroom, Harry Potter stirs from a restless sleep. He gropes for his
glasses. She goes to Ron. Harry rubs the jagged scar on his forehead. Later, the
kids walk through a forest with Ron's father, his sister Ginny; and twin brothers
Fred and George. A stout, smiling man hikes toward them.

Segment of interest: In a bedroom, Harry Potter stirs from a restless sleep.

[{"question": "Who rubs their forehead?", "correct_answer": "Harry",
"wrong_answers": ["the caretaker", "Ron", "Ginny", "Fred"]}, {"question": "What
shape is the scar on Harry's forehead?", "correct_answer": "jagged",
"wrong_answers": ["circular", "crescent", "oval", "straight"]}, {"question":
"Where on Harry's body is there a jagged scar?", "correct_answer": "forehead",
"wrong_answers": ["cheek", "neck", "arm", "calf"]}]

Your task

Full transcript: INSERT_FULL_TRANSCRIPT

Segment of interest: INSERT_SEGMENT_OF_INTEREST
```

We filtered the resulting questions aggressively, keeping those that: pass QEval on both reference scripts, fail QEval on the same script without the segment on which the question

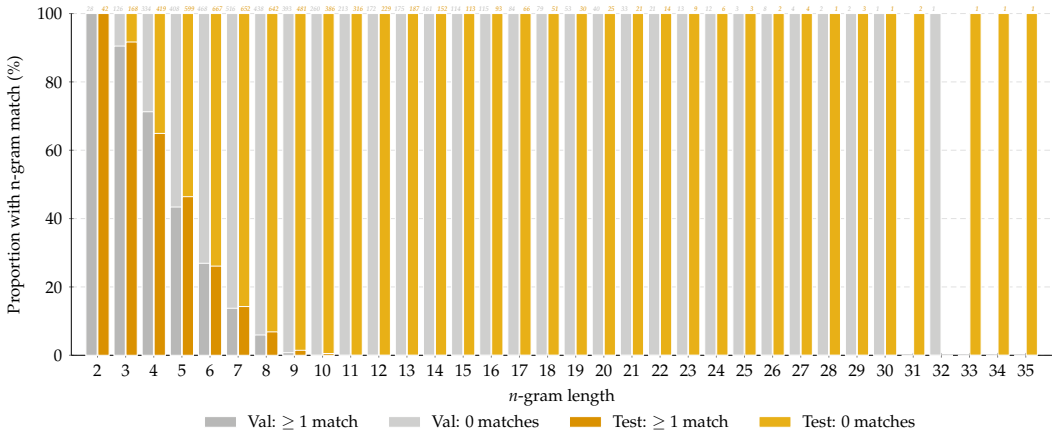


Figure 8: Proportion of *AD* segments with at least one exact n-gram match in the LLM evaluator's training corpus across validation and test splits. The numerical values above each bar indicate the absolute count of segments evaluated at that specific length. The distribution reaches zero matches for segments of length greater than 10 tokens, which is below the 13-gram threshold set by Brown et al. (2020), suggesting very low contamination of our AD evaluation data.

was based, pass QEval-T on both reference scripts. Table 13 gives a summary of our filtering stages and how many QA pairs remain after each stage.

Stage	description	test QAs	challenge QAs
0	All generated questions	14,981	58,681
1	Pass QEval on the source reference	13,737	53,704
2	Fail QEval on the source reference without the source segment	5,042	20,829
3	Pass QEval on the alternative reference	2,182	8,335
4	Pass QEval-T on the source reference	1,987	7,715
5	Pass QEval-T on the alternative reference	1,517	5,560

Table 13: Number of QA pairs remaining after each filtering stage.

QA answering is performed with a base LLM using a basic prompt template identical to the multiple-choice data used in the model's mid-training.[8] We compare the resulting space-prefixed A-E logits.

We access this LLM evaluator's 6T token training corpus for contamination of the textual AD using the infini-gram tool (Liu et al., 2024). Results show zero exact matches against AD segments containing greater than 10 tokens, leading us to determine the risk of contamination is low (Figure 8; Figure 9 provides a reference analysis of dialogue subtitles).

[8] `{doc}\nQuestion: {question}\nA. {answer_a}\n...\nE. {answer_e}\nAnswer:`

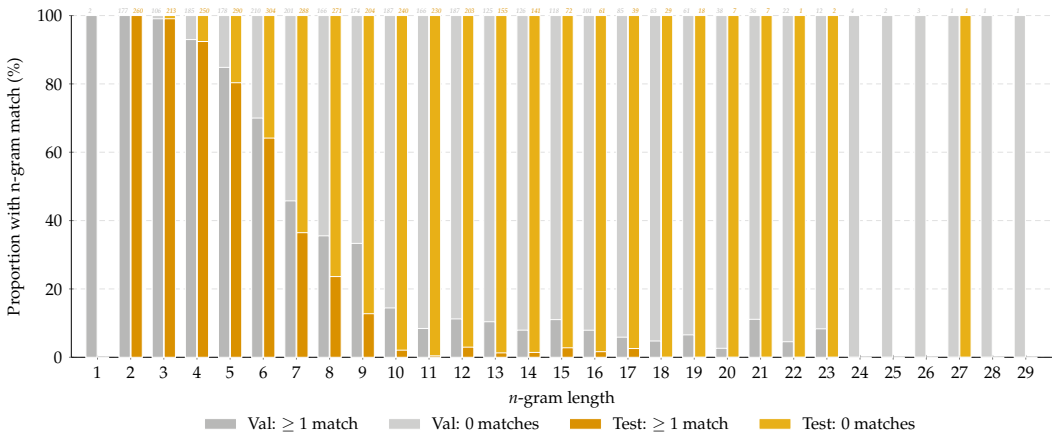


Figure 9: Proportion of *dialogue subtitle* segments with at least one exact n-gram match (a direct match or match with newlines replaced with spaces) in the LLM evaluator's training corpus across validation and test splits. The numerical values above each bar indicate the absolute count of segments evaluated at that specific length. In both splits, we find matches for grams up to length 17 tokens, demonstrating contamination. The contamination is noticeably higher for the validation set as compared to the test set (which contains newer movies from 2023–2025).

M LLM AD Generation

M.1 Prompt

LLM Prompt

```
Task description:

You are provided with the video (no audio) and dialogue subtitles for a full movie.
Your task is to generate an audio description script for one part of the movie.

Audio description is a verbal narration of key visual content in a video.
Descriptions must be inserted into gaps between dialogue.

Output format:

Return exactly one valid JSON object and no other text. Each key represents a gap
between dialogue. Each value is an array of description objects. Each description
object contains a textual description and timestamps for when it should be
narrated. The following illustrates the required structure:

{"1st DIALOGUE GAP START --> DIALOGUE GAP END": [{"text": "TEXT", "start":
"START", "end": "END"}, {"text": "TEXT", "start": "START", "end": "END"}, {"text":
"TEXT", "start": "START", "end": "END"}], "2nd DIALOGUE GAP START --> DIALOGUE GAP
END": [{"text": "TEXT", "start": "START", "end": "END"}]}

- DIALOGUE GAP START --> DIALOGUE GAP END takes the form "TIMESTAMP_FORMAT -->
TIMESTAMP_FORMAT" and is the temporal interval when descriptions can be inserted.
- TEXT is a textual description.
- START and END are timestamps with format "TIMESTAMP_FORMAT".

Rules:
```

```
- Each value must be an array of zero, one, or more description objects, each
containing exactly the keys "text", "start", and "end".
- For each description object:
  - TEXT is one sentence or shorter
  - # words in TEXT approx 3 x (END - START) in seconds
  - START >= DIALOGUE GAP START
  - END <= DIALOGUE GAP END
  - END > START
- Description objects must never overlap in time

Audio description script for prior parts:

INSERT_PRIOR_AUDIO_DESCRIPTION

Required part:

You should now generate an audio description script for the part of the movie that
starts at PART_START_TIME and ends at PART_END_TIME.

Gaps between dialogue in the required part of the movie:

The following is the required output template, with the dialogue gaps in the
required part of the movie included as keys:

OUTPUT_TEMPLATE

Generate the JSON with the values filled in. Your JSON must include all keys in
the template above.
```

For chunked video inputs, we adapt the prompt to state that the supplied video is one part of the movie rather than the full movie, remove the absolute start and end times of the required part, and express the subtitles, dialogue gaps, and generated timestamps relative to the beginning of the chunk.

Video (1FPS, no audio) is the first part of every prompt, followed by dialogue subtitles. Dialogue subtitles take the form `START --> END TEXT`. For dialogue gaps, we include those that are at least one second in duration. Timestamps are formatted according to model video preprocessing (`MM:SS` for Gemini and `<S.S seconds>` for Qwen). For chunked video generation, relative output timestamps are converted to absolute movie times.

Iterative prompts include previously generated descriptions that were successfully parsed and validated against the prompt instructions:

- Malformed JSON (between first and last braces) and responses terminating due to maximum output length are treated as empty output.
- We retain only keys that match an eligible dialogue gap and whose values are arrays. For Qwen, we additionally accept otherwise exact gap keys from which the angle brackets around timestamps are missing.
- We retain only description objects containing the required `text`, `start`, and `end` fields, with nonempty textual descriptions and parseable timestamps.
- We require each segment to have a strictly positive duration and to be fully contained within the dialogue gap identified by its key.
- Within each gap, we remove duplicate segments, order the remaining segments by start and end time, and discard any segment that overlaps the preceding retained segment.

For chunked video input with iterative generation on our challenge set, timestamps risk becoming ambiguous. From the second ten-minute chunk onwards, timestamps in the accumulated AD script refer to the movie timeline, while those in the required output refer to the current chunk's timeline. To address this ambiguity, we compared providing the AD script for prior chunks as prose text, without timestamps, against providing it in the script

format. Results are in Table 14. The two prompting approaches are largely indistinguishable, with small improvements in the QA-based metrics for the prose variant, which we therefore adopt.

	CIDEr	METEOR	SODA-M	SODA-T	QEval	QEval-T
Qwen prose	13.2 (0–2)	8.1 (0–0)	7.4 (0–2)	38.4 (2–0)	43.9 (3–0)	17.3 (3–0)
Qwen script	13.3 (0–2)	7.8 (0–0)	7.4 (0–2)	35.4 (0–1)	43.4 (2–1)	16.7 (1–1)
Gemini prose	19.0 (2–0)	8.2 (0–0)	7.9 (2–0)	36.0 (0–0)	42.3 (1–2)	16.4 (1–1)
Gemini script	19.6 (2–0)	8.2 (0–0)	7.9 (2–0)	34.6 (0–1)	40.9 (0–3)	15.8 (0–3)

Table 14: Iterative prompting strategy for chunked video input on our challenge set. Numbers in brackets represent wins and losses in the significance tests for that metric (W–L).

M.2 Inference Configuration

Table 15 shows our inference configurations used with the Qwen 3.5 27B and Gemini 3.1 Flash-Lite LLMs.

Parameter	**Gemini 3.1 Flash-Lite**	**Qwen3.5-27B**
Temperature	1.0	1.0
Top-p	0.95	0.95
Top-k	64	20
Min-p	–	0.0
Presence penalty	–	1.5
Repetition penalty	–	1.0
Max output tokens	65 536	81 920
Thinking	`high`	enabled
Frame rate (FPS)	1	1
`longest_edge`	–	469 762 048
`shortest_edge`	–	4 096
RoPE type	–	YaRN (θ=10^7, factor=4)

Table 15: Inference parameters for each LLM.

N Complementary Results

	CIDEr	METEOR	SODA-M	SODA-T	QEval	QEval-T
Expert	51.4	20.1	16.3	81.0	69.6	61.2
Random	1.3 (1–0)	4.4 (1–0)	4.0 (1–0)	27.1 (1–0)	34.1 (0–1)	2.3 (1–0)
Empty	0.0 (0–1)	0.0 (0–1)	0.0 (0–1)	0.0 (0–1)	41.4 (1–0)	0.0 (0–1)
Gemini (c)	19.0 (4–0)	8.2 (4–0)	7.9 (5–0)	36.0 (3–0)	42.3 (3–1)	16.4 (4–1)
Gemini (c, non-it.)	18.1 (4–0)	7.9 (3–1)	7.7 (3–1)	34.5 (3–1)	42.0 (3–1)	15.7 (3–2)
Gemini	12.2 (1–2)	6.1 (1–3)	6.8 (1–3)	27.1 (1–3)	40.1 (0–4)	10.6 (2–3)
Gemini (non-it.)	10.5 (0–4)	5.6 (0–4)	6.5 (0–4)	23.4 (0–4)	40.4 (1–3)	9.4 (0–4)
Qwen (c)	13.2 (2–2)	8.1 (3–0)	7.4 (3–1)	38.4 (4–0)	43.9 (5–0)	17.3 (5–0)
Qwen	10.3 (0–3)	6.1 (0–3)	6.8 (0–3)	26.0 (0–3)	40.0 (0–3)	9.4 (0–4)
NarrAD (w/o cur.)	13.3 (0–4)	7.9 (6–0)	8.5 (3–0)	55.0 (5–0)	45.5 (6–0)	19.1 (6–0)
NarrAD	14.6 (1–0)	7.3 (1–1)	8.5 (3–0)	49.7 (1–2)	43.3 (5–1)	18.7 (5–1)
ShotbyShot (GPT4o)	16.3 (2–0)	7.0 (1–1)	8.0 (1–3)	53.0 (5–0)	39.9 (4–2)	17.0 (4–2)
UniAD	15.9 (2–0)	7.3 (2–1)	8.2 (1–0)	48.5 (0–2)	36.8 (3–3)	11.3 (2–3)
DistinctAD	15.6 (2–0)	7.1 (1–1)	8.3 (3–0)	49.0 (0–2)	35.7 (1–4)	8.8 (0–6)
ShotbyShot (Q/L)	14.8 (1–0)	6.9 (1–2)	8.0 (1–3)	46.7 (0–3)	35.8 (1–4)	11.5 (2–3)
AutoAD Zero	13.4 (0–4)	6.5 (0–6)	7.6 (0–6)	47.1 (0–2)	34.7 (0–6)	9.8 (1–5)

Table 16: Challenge set system performance under the six evaluation metrics. Numbers in brackets represent wins and losses in the significance tests for that metric (W–L), within a category. c=chunk, it.=iterative, cur.=curation, Q/L=Qwen/Llama.

	CIDEr	METEOR	SODA-M	SODA-T	QEval	QEval-T
Gemini 3.1 w/ faces	22.5 (4–0)	9.7 (3–0)	8.9 (4–0)	52.2 (1–0)	44.7 (1–0)	23.1 (1–0)
Qwen 3.5 w/ faces	18.0 (2–1)	9.6 (3–0)	8.2 (2–1)	55.7 (2–0)	43.8 (1–0)	21.7 (1–0)
Qwen 3.5	12.8 (1–2)	7.4 (1–2)	6.7 (1–2)	54.0 (2–0)	42.3 (1–0)	20.2 (1–0)
Gemini 3.1	14.5 (1–1)	7.3 (1–2)	7.3 (1–1)	49.6 (1–2)	42.2 (1–0)	21.9 (1–0)
Random	1.1 (0–4)	3.9 (0–4)	3.8 (0–4)	39.5 (0–4)	32.4 (0–4)	7.8 (0–4)

Table 17: Test set system performance under the six evaluation metrics. Numbers in brackets represent wins and losses in the significance tests for that metric (W–L). The systems are sorted by the sum of net wins across all metrics.

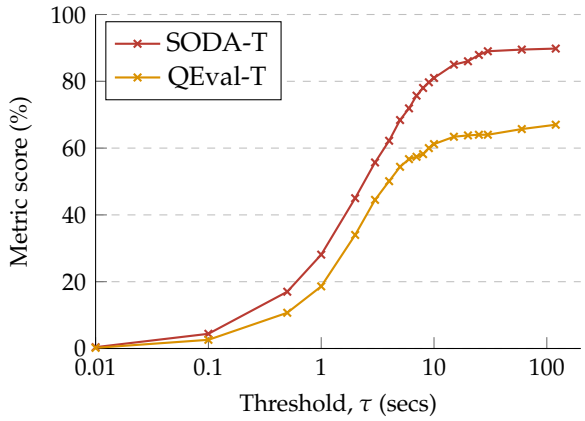


Figure 10: Sweep of the threshold used in the SODA-T and QEval-T evaluation metrics. Results based on the expert human upper bound on the challenge set. The metrics plateau with scores lower than 100%: SODA-T relies on alignment and unaligned description elements never score; QEval-T relies also on QA answer correctness, so tends towards QEval.

Gap length	CIDEr					METEOR				
	Expert	Qwen 3.5		Gemini 3.1		Expert	Qwen 3.5		Gemini 3.1	
		Full	Chunk	Full	Chunk		Full	Chunk	Full	Chunk
1–5s	57.1	10.1	12.0	12.4	19.7	18.3	5.8	7.3	5.7	8.0
5–10s	43.5	11.0	17.4	13.0	22.3	19.2	6.5	8.9	6.6	8.9
10–20s	50.0	5.4	11.2	7.4	11.7	21.9	5.8	8.7	6.6	8.8
20–30s	41.6	1.5	6.6	2.0	3.8	20.8	4.9	8.6	5.6	8.1
30–60s	22.6	0.8	2.0	1.7	1.2	21.4	5.3	8.2	5.7	7.6
>60s	23.3	1.4	1.2	0.0	0.4	22.1	5.3	8.5	5.6	6.8

Table 18: CIDEr and METEOR-based performance across varying dialogue gap lengths. Note the drop in CIDEr scores for dialogue gaps of length greater than 20 seconds, which is due to a length-difference penalty term in the metric. Note also particularly good Gemini performance for short gaps.

System	METEOR		SODA-M		SODA-T		QEval		QEval-T	
	US	UK	US	UK	US	UK	US	UK	US	UK
Expert	16.5	18.0	14.4	15.5	80.4	73.0	69.0	70.0	60.0	62.3
Qwen 3.5	4.8	5.3	6.0	6.6	20.9	21.8	39.6	40.3	8.8	9.8
Qwen 3.5 (chunk)	6.7	7.2	6.7	7.1	32.5	33.4	43.6	44.4	16.7	17.9
Gemini 3.1	4.9	5.2	6.0	6.6	21.8	22.8	40.2	40.2	11.0	10.3
Gemini 3.1 (chunk)	6.7	7.0	7.0	7.6	29.8	31.5	42.5	42.3	17.0	15.9

Table 19: Downstream performance against each of our two references (US and UK) individually. CIDEr results cannot be reported here as it relies on both references jointly.

	Words per minute		
	Full movie	Chunk	Test
US reference	196.2	—	195.0
UK reference	201.5	—	197.6
Gemini 3.1	110.3	118.3	157.0
Qwen 3.5	104.9	145.0	182.9

Table 20: Speech rate for professional AD references and LLM generations. References are tightly centred around roughly 200 words per minute, close to guideline recommendations. In contrast, Gemini and Qwen show inconsistent timing behaviour: with increasing length video input, generations become overly brief. On the test set of shorter videos, the models generate approximately three words per second, in accordance with their instructions.