Synopses of Movie Narratives: a Video-Language Dataset for Story Understanding

Anonymous ACL submission

Abstract

Despite recent advances of AI, story understanding remains an open and under-investigated problem. We collect, preprocess, and publicly release a video-language story dataset, Synopses of Movie Narratives (SyMoN), containing 5,193 video summaries of popular movies and TV series. SyMoN captures naturalistic storytelling videos for human audience made by human creators, and has higher story coverage and more frequent mental-state references than similar video-language story datasets. Differing from most existing video-text datasets, SyMoN features large semantic gaps between the visual and the textual modalities due to the prevalence of reporting bias and mental state descriptions. We establish benchmarks on video-text retrieval and zero-shot alignment on movie summary videos. With SyMoN, we hope to lay the groundwork for progress in multimodal story understanding.

1 Introduction

Stories are complex artifacts that succinctly encode the human experience. The understanding of story content involves high-level semantic concepts such as character motivations and intentions (Emelin et al., 2020; Rashkin et al., 2018), events structures (Chambers and Jurafsky, 2008; Li et al., 2013; Pichotta and Mooney, 2016; Ferraro and Van Durme, 2016; Martin et al., 2018; Wang et al., 2021; Caselli et al., 2021), as well as social relationships among story characters (Elson et al., 2010; Chaturvedi et al., 2016; Kim and Klinger, 2019). To this day, understanding of story semantics remains an open and under-investigated problem.

The recent emergence of user-generated, "a movie in X minutes" videos offers a rich source of naturalistic storytelling videos. These videos usually select clips that depict key story events from a movie or a TV series. The narrator recounts the story alongside the video. These videos provide condensed yet complete storylines that are carefully assembled for human viewers by human creators.

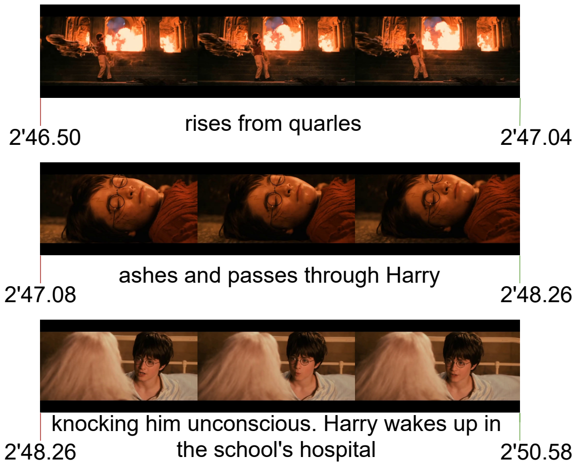

Figure 1: An example video with narration text from SyMoN. The video has been automatically segmented into three scenes. We show the boundary timestamps.

We identify, collect, preprocess, and publicly release a video-language story dataset, named Synopses of Movie Narratives (SyMoN). The dataset includes 5,193 user-generated video summaries of popular movies and TV series for a total length of 869 hours. For 857 movies, multiple summary videos are available, which may be used as references for generation or summarization. In Figure 1, we show an example video and text description from SyMoN. We empirically verify SyMoN as the prototypical story dataset, as it has higher coverage of plotlines and more frequent mental-state references than several similar video-language story datasets.

However, the nature of storytelling poses unique obstacles for computational understanding due to the semantic divergence between the video and text. First, in the phenomenon known as reporting bias (Gordon and Van Durme, 2013), human narrators tend to avoid stating the obvious. For example, in Figure 1, the video shows Harry Potter lying on the floor, while the narrator states "... knocking him unconscious". To recognize that lying on the floor is a consequence of being knocked unconscious requires event-level cause-and-effect reasoning, which may prove difficult for today's AI (Sap et al., 2019). Second, the story texts contain frequent mentions of story characters' mental states (§5.2), which may not be easily recognizable from video. This contrasts with crowdsourced datasets like Charades (Sigurdsson et al., 2016) where humans are asked to follow textual instructions, or LSMDC (Rohrbach et al., 2017) where the narration meticulously describes the imagery for audience with visual impairment.

To examine the cross-modality semantic gap, we design a simple task that temporally orders two video segments. A large pretrained UniVL model (Luo et al., 2020) demonstrates mediocre performance and limited utilization of textual information, highlighting the challenge posed by SYMON.

As benchmarks for future research, we establish baselines for text-to-video and video-to-text retrieval on SYMON and a zero-shot video-text alignment baseline using the YMS dataset as test. Together, the weakly supervised SYMON and the fully annotated YMS form a complete benchmark, serving as a new challenge for the multimodal research community.

Our contributions are three-fold:

- We collect, preprocess, and publish a large-scale movie summary dataset, which can support various multimodal tasks such as retrieval, captioning, and summarization.
- We preform extensive experiments to quantify the characteristics of SYMON, including its coverage of major plotlines, the amount of mental-state descriptions, and the semantic divergence between text and video.
- To facilitate future research, we establish baselines for text-video retrieval on SYMON and zero-shot transfer to the YouTube Movie Summary dataset (YMS) (Dogan et al., 2018).

2 Related Work

Datasets for Event and Story Understanding. Events and story structures are closely related (Caselli et al., 2021). Existing datasets provided annotations for the the temporal aspects, such as temporal precedence and duration (UzZaman et al., 2013; Chambers et al., 2014; Ning et al., 2020; Zhou et al., 2021; Vashishtha et al., 2019, 2020), and causal relations between events (O'Gorman et al., 2016; Roemmele et al., 2011).

Several datasets explore individual components of stories, including sentence ordering (Gangal et al., 2021), social norms and moral consequences (Emelin et al., 2020), plausible antecedent (Bhagavatula et al., 2020), intentions and effects on mental states (Rashkin et al., 2018), high-level story structures (Ouyang and McKeown, 2015; Li et al., 2018), and story character descriptions (Brahman et al., 2021). Sap et al. (2019) consider relations between events, persona, and mental states. Some datasets aim at summarization for screenplays or conversation transcripts (Gorinski and Lapata, 2015; Papalampidi et al., 2020; Chen et al., 2021). Notably, Sadhu et al. (2021) annotate event relations from video.

Researchers also develop general-purpose QA datasets conditioned on comprehension of story texts, such as MCTest (Richardson et al., 2013), NarrativeQA (Kočiský et al., 2018), and FriendsQA (Yang and Choi, 2019). Multimodal counterparts like MovieQA (Tapaswi et al., 2016), TVQA (Lei et al., 2018), and Pororo (Kim et al., 2017) are available. However, not every question in the QA datasets requires in-depth narrative understanding.

Video-Text Movie Story Datasets. A number of datasets supply story content extracted from movies. The Large-Scale Movie Description Challenge (LSMDC) (Rohrbach et al., 2017) combined the efforts of MPII-MD (Rohrbach et al., 2015) and M-VAD (Torabi et al., 2015) and provide detailed language descriptions initially intended for the visual impaired. Although these descriptions are highly accurate, they may not be representative of real-world storytelling.

YouTube Movie Summary (YMS) (Dogan et al., 2018) contains 94 YouTube movie summary videos with human-narrated storylines. The Condensed Movies Dataset (CMD) (Bain et al., 2020) gathers 7 to 11 key clips from each movie with one-sentence descriptions for each clip. Pororo (Kim et al., 2017) captures 20-minute cartoon episodes, in-show conversations, and human-written descriptions. MovieNet (Huang et al., 2020) annotate 2000 hours of movies with extensive annotations and aligned movies scripts. However, due to copyright, legal clearance for the video release is still pending at the time of writing.

	Video hours	#Videos (#Clips)	#Sent	Vocab.
CMD	1,270	3,605 (33,976)	35,681	15,272
MovieNet (video release pending)	2,000	1,100		
LSMDC	147	200 (128,085)	128,118	22,500
Pororo	20.5	171 (16,066)	43,394	
MovieGraph	94.0	51 (7,637)	20,849	
SyMoN (Ours)	869	5,193	683,611	40,116

Table 1: Comparison of video description datasets with story content.

Other types of video annotations have been explored, including semantic roles and event relations (Sadhu et al., 2021), character relationships and types of speech (Wu and Krahenbuhl, 2021), and movie graphs (Vicol et al., 2018).

In this work, we collect a large-scale, readily available, multi-reference dataset of human-curated movie summaries, named SyMoN. The dataset can be leveraged for various story understanding and generation tasks such as sequential text localization, story generation from video, and movie summarization. To our knowledge, SyMoN is the largest dataset for short naturalistic storytelling videos.

3 Dataset Collection and Statistics

We apply the following procedure for data collection. First, we manually identify relevant YouTube channels by searching with keywords such as "movie summary", "movie recap", and "movie shortened". We download all videos from the identified channels and accompanying subtitles, which may be written by humans or automatically generated by YouTube. Videos without subtitles are excluded. Finally, we perform rule-based extraction of movie names from metadata and subtitles and discard videos that are not movie summaries.

This yields a total of 5,193 videos with an average length of 9.5 minutes and a total length of 869 hours. On average, the narration in one video contains 1,717 words or 131 sentences. The overall vocabulary size is 40,116. SyMoN contains summaries for 2,440 movies and TV series, of which 857 have more than 1 summary. The most popular TV series, *The Walking Dead*, has 84 summaries. On average, one movie or TV series in the 857 has 4.21 summaries. Compared to existing datasets (see Table 1) SyMoN is one of the largest movie narrative datasets with most diverse vocabulary. In addition, SyMoN has more complete coverage of story events than LSMDC and CMD (§5.1).

4 Preprocessing

Subtitle Masking. Some videos have subtitles embedded in the video. In tasks like text-to-video retrieval, the embedded subtitles may become a shortcut feature, causing networks to learn only optical character recognition.

To eliminate shortcuts, we locate embedded subtitles and mask them out. For efficiency, we randomly sample 100 frames from each video and apply an accurate text detection technique (Baek et al., 2019). Observing that the subtitles are almost always at the same location in a single video, we take the minimum bounding box that can cover all embedded subtitles in all 100 frames as the masked region; we set all pixels in the region to black.

Punctuation Restoration. We acquire subtitle texts from YouTube directly. Sometimes the texts are the result of automatic speech recognition, which cannot recognize punctuation. To fix this, for every unpunctuated narration text, we generate punctuation with (Alam et al., 2020).

Scene Segmentation. Later experiments require temporal segmentation of videos based on camera cuts. For this purpose, we run the dataset through the network of Souček and Lokoč (2020), which detects hard camera cuts. A scene, defined as the continuous shot between two cuts, lasts 2.2 seconds on average. This is similar to CMD, another movie dataset, whose scenes last 2.4 seconds on average. However, average scenes in ActivityNet (Caba Heilbron et al., 2015) and Kinetics-400 (Kay et al., 2017) last for 11.1 seconds and 30 seconds respectively. This shows camera cuts in movies are much more frequent than the user-generated videos in ActivityNet and Kinetics.

5 Characteristics of SyMoN Stories

5.1 Story Coverage

To facilitate story understanding, it is desirable that, despite their short lengths, the videos in SyMoN provide sufficient coverage (Bain et al., 2020) over major plot points of the original movies or TV shows. In this experiment, we treat Wikipedia plot summaries (WikiPlots)[1] as ground truth and estimate the extent the stories in CMD, LSMDC, and SyMoN cover the sentences in WikiPlots.

We use a three-step procedure for computing story coverage. First, we match movie summary in our dataset to their WikiPlots summaries by name. Second, we estimate if a sentence from the video narration is equivalent to a sentence in WikiPlots using the natural language inference (NLI) classifier from Nie et al. (2020). From two input sentences a and b, the NLI classifier predicts one of three possibilities: a entails b; a contradicts b; and neither is true. As entailment is asymmetric, we use the average probability for both directions (a entails b and b entails a) as the probability that a and b are equivalent. Finally, we find the best correspondence between two texts using Dynamic Time Warping (DTW) (Berndt and Clifford, 1994), which optimizes correspondence over entire sequences.

Briefly, DTW is a dynamic programming algorithm that seeks minimum-cost correspondence between two sequences, the WikiPlots sentence sequence A, and the narration sentence sequence B. We refer readers to the Appendix for a detailed description of the DTW algorithm. Using manually labeled sentence correspondences, we determine two model parameters, δ_A and δ_B, which denote the respective costs for skipping a sentence in sequences A and B.

We manually labeled the correspondence between around 500 sentences in CMD with Wikiplots stories, and did the same for SyMoN. For LSMDC, we labeled around 1300 sentences because LSMDC texts are much longer. A second annotator labeled a small portion of data from each dataset to compute inter-rater reliability. The Cohen Kappa on SyMoN, CMD and LSMDC are 0.86, 0.59, and 0.33 respectively. We believe the low agreement on LSMDC is caused by the mismatch in the text lengths. Texts in LSMDC are longer than all other story texts, which led to difficulties in precisely locating the correspondence.

[1] https://github.com/markriedl/WikiPlots

	CMD	LSMDC	SyMoN
Story Coverage	10.8%	18.1%	37.9%

Table 2: Estimated story coverage with sentence entailment and Dynamic Time Warping.

With a grid search, we find the optimal δ_A and δ_B as those that cause DTW to identify matched sentences the most accurately. The accuracy is defined as

$$\text{Accuracy} = \frac{1}{2}\left(\frac{T^{\text{wiki}}}{N^{\text{wiki}}} + \frac{T^{\text{text}}}{N^{\text{text}}}\right). \qquad (1)$$

Here T^{wiki} and N^{wiki} are the number of correctly matched and the total number of WikiPlots sentences, respectively. T^{text} and N^{text} are the number of correctly matched and the total number of video narration sentences. We do not directly optimize story coverage because doing so results in incorrectly matched sentences that artificially inflate the story coverage measurement.

With the optimal δ_A and δ_B, we perform DTW again and calculate story coverage as the proportion of WikiPlots sentences matched with narration sentences,

$$\text{Coverage} = \frac{1}{K}\sum_{i}^{K} \frac{M_i}{N_i^{\text{wiki}}}, \qquad (2)$$

where K is the number of WikiPlots movies appearing in the video dataset. In the i^{th} WikiPlots text, M_i denotes the number of matched sentences and N_i^{wiki} denotes the total number of sentences.

Table 2 shows the story coverage results. Of the three datasets, SyMoN provides the highest coverage. LSMDC comes in second place, partially because it contains significantly longer descriptions for each movie than the other datasets.

5.2 Mental State Descriptions

A crucial component of story understanding is to develop theory of mind for the story characters, that is, to understand their mental states, such as emotions, motivations, and intentions (Bruner, 1986; Happé, 1994; Pelletier and Beatty, 2015). However, these concepts tend to be under-represented

	Emotion	Motivation	Intention
CMD	38.9	1.41	9.4
LSMDC	33.5	0.62	2.8
AcitivityNet Captions	27.5	0.51	2.7
SYMON (Ours)	**57.6**	**1.58**	**23.9**

Table 3: Frequency of words related to emotion, motivation, and intention per one thousand words in the text corpora.

in the textual descriptions from commonly used video-language datasets.

In this experiment, we measure the frequency of words related to emotions, motivations, and intentions in the text associated with the videos. For emotional words, we adopt the WordNet-feelings dataset (Siddharthan et al., 2018), which includes 11387 emotion-related words identified by human experts. For motivation and intention words, we find 200 nearest neighbors of the words "motivation" and "intention" using 300-dimensional fastText embedding (Bojanowski et al., 2017) trained on Wikipedia and Common Crawl[2]. We select 200 words as we find additional neighbors to be irrelevant to motivation and intention.

Table 3 reports word frequencies for every thousand words in four video-language datasets. We observe that SYMON employs mental-state words the most frequently and uses intention-related words 2.5 times as often as the next dataset, CMD. ActivityNet Captions (Krishna et al., 2017), containing matter-of-fact descriptions of actions in generic user-uploaded videos, uses the least of such words. LSMDC, which contains literal descriptions of movie clips, is ranked the third. CMD has a focus on the story content and is ranked the second. Overall, we find the ranking consistent with the nature of the datasets, as story text describes mental states more often than literal descriptions of generic videos. SYMON appears to be the most prototypical story dataset of the four.

6 Understanding Video-Text Divergence by Sequencing Videos

As discussed earlier, SYMON are characterized by large gaps between the textual and visual modalities due to the reporting bias, or the tendency to avoid stating what can be easily observed from the video, and the prevalence of mental state descriptions, which are often not visible from the video. In this section, we report an experiment designed to estimate the extent of video-text correspondence.

Problem Definition. Similar to event/sentence ordering (Liu et al., 2018; Devlin et al., 2019), we predict the correct ordering of two consecutive video segments separated by a hard camera cut. The network predicts one of two classes: video segment 1 precedes segment 2 or vice versa. To create balanced classification, we randomly flip the ordering of the two video segments. We extract the text description that spans the same duration as the two video segments and expand the text to sentence boundaries.

We design two networks, one utilizing the unaltered textual description and the other solely relying on visual input. This setup allows us to estimate the amount of information provided by text. That is, if the text provides grounding to elements in both video segments, it should help the text-aware network predict the correct ordering.

Network Architecture. We adopt three pretrained modules, the text encoder, the video encoder, and the cross-modality encoder from UniVL (Luo et al., 2020), which are pretrained on HowTo100M (Miech et al., 2019), and finetune the weights. The two video segments are encoded separately and their features are concatenated with the encoded text feature. After that, the two groups of features go through the cross encoder independently, yielding feature vectors $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$. With parameter $\boldsymbol{w}$, the prediction is

$$P(\hat{y}=1) = \sigma(\boldsymbol{w}^\top \boldsymbol{f}_1 - \boldsymbol{w}^\top \boldsymbol{f}_2). \qquad (3)$$

where $\sigma(\cdot)$ is the sigmoid function and $\hat{y}$ is the predicted class index. Figure 2 shows the overall network architecture.

As a baseline, we also create a network that relies on only the visual input, in which we replace the textual feature fed into the cross encoder with an all-zero vector. The rest of the network architecture remains the same.

Experimental Setup. To cover as much data as possible, we adopt a special dataset split, containing Set A of 2,444 videos, Set B of 2,289 videos, and a validation set of 500 videos. Each network is trained on Set A and tested on Set B, and then trained on Set B and tested on A. We report the

[2] Acquired from https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md.

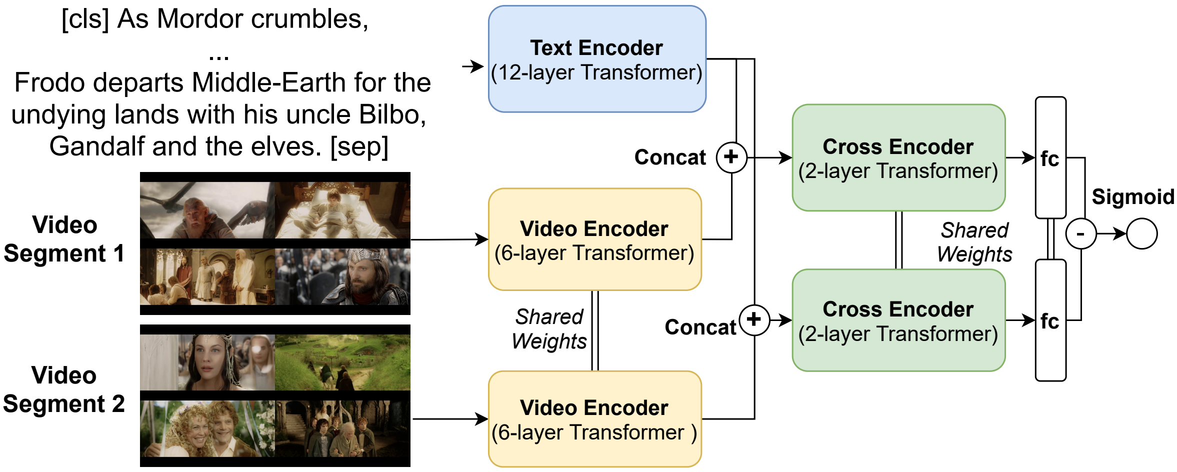

Figure 2: The network architecture for the temporal ordering task. The double vertical lines indicate weight sharing between modules.

average test accuracy. We tuned hyperparameters extensively on the validation set and select the training epoch with the highest validation accuracy. To avoid test data leak, we put all videos of the same movie or movie franchise to the same set. More settings can be found in the Appendix.

Results and Discussion. Table 4 lists the predictive accuracy. The network based solely on video has an accuracy of 63.4%. The incorporation of textual information improves prediction accuracy by 5.7%. Noting that chance level is at 50%, we find the performance to be mediocre. Since UniVL has been pretrained on HowTo100M and provides a good initialization, the results underscore the effects of the semantic gap between video and text.

Without text, 36.6% of data points cannot be correctly sequenced. Out of these, $5.7/36.6 = 15.6\%$ can be correctly classified with text. As the 36.6% are difficult data samples, we estimate the probability that (1) the text makes reference to both video segments *and* (2) the network correctly recognizes the references to be *at least* 15.6%.

Data Samples. In Figure 3, we present two data points, one from the 5% most helpful text cluster and one from the 5% least helpful text cluster. We observe that the helpful text mentions objects such as cauldron and book that appear in both video segments. As a result, both video segments can be grounded in the text, which provides ordering information. In comparison, the unhelpful text mentions rare object and action such as cat costume and jewelry robbery, which are difficult for the network to learn. Similarly, connecting the text "the mother refuses her son" and the discussion shown in video is not straightforward and would require identity tracking and event understanding.

	Text + Video	Only Video
Accuracy	69.1%	63.4%

Table 4: Temporal order prediction accuracy of the text-aware and visual-only models.

Object and Action Analysis. We examine the match between text and video with contemporary technology on object detection and action recognition. First, for every data point, we compute the confidence of the ground-truth class from the two models. If the text-aware model has higher confidence than the visual-only model, we consider the text to be helpful. We rank the data points by the confidence difference between the two models, and take 5% data with the most helpful text and 5% data with the least helpful text.

Next, we run Faster-RCNN (Girshick, 2015) trained on Open Images V4 (Kuznetsova et al., 2020) to detect 600 object classes on video frames, and 3D-ResNet (Hara et al., 2018) trained on Kinetics-700 (Kay et al., 2017) to detect 700 action classes. After that, we match the identified objects and actions to the texts. The Appendix contains more details.

Table 5 shows that the most helpful texts contain relatively 18.8% more recognizable objects and 25.0% more actions than the most unhelpful texts. This suggests that textual references to ob-

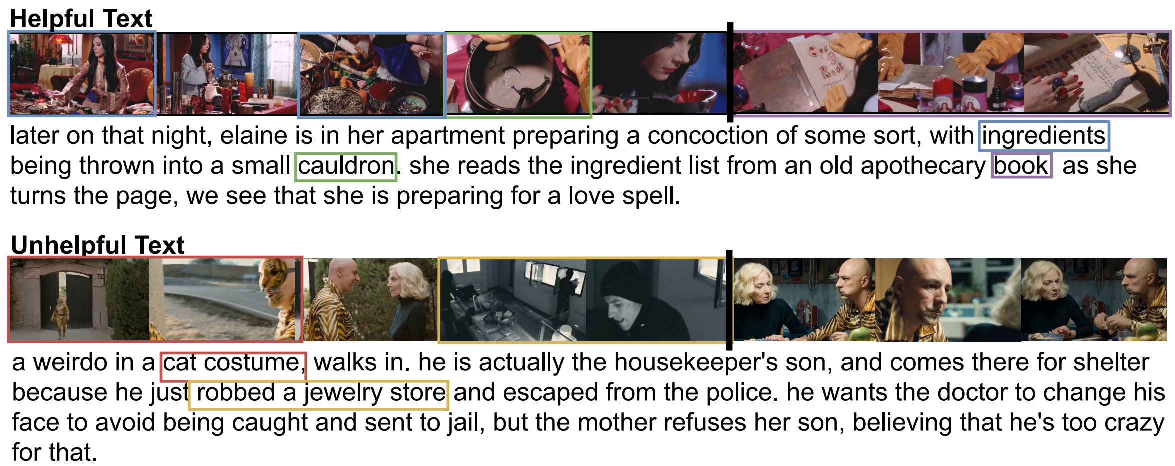

Figure 3: Examples from the most and least helpful text clusters. Bound boxes of the same color in text and video frame denote video-text correspondence. The black line denotes the boundary between the two video segments to be ordered.

	Objects Detected	Actions Recognized
Helpful Text	0.19	0.20
Unhelpful Text	0.16	0.16

Table 5: Number of words that match exactly the detected object names or action names per text description.

jects and actions in the video may have contributed to the temporal ordering task. Noting that a text description in this experiment contains 83 words on average, the detected objects and actions appear rather scarce. We once again attribute this observation to the reporting bias in the dataset.

7 Multimodal Retrieval

In this section, we establish baselines on the task of video-text retrieval on SYMON and the YouTube Movie Summary (YMS) (Dogan et al., 2018), which serve as benchmarks for future research.

7.1 Network Architecture

We employ pretrained UniVL encoders without the cross encoder. We encode the i^{th} text with the text encoder, producing feature $\boldsymbol{t}_i$, and encode the i^{th} video segment with the video encoder, producing feature $\boldsymbol{v}_i$. The similarity between the two is simply their dot product. With randomly sampled negative text features $\boldsymbol{t}_k, k \neq i$ and video features $\boldsymbol{v}_k, k \neq i$, we use the NCELoss (Gutmann and Hyvärinen, 2010):

$$L_{\text{NCE}} = \frac{1}{N}\sum_{i=1}^{N} -\boldsymbol{v}_i^\top \boldsymbol{t}_i + \log\Big(\exp \boldsymbol{v}_i^\top \boldsymbol{t}_i + \sum_{k\neq i}^{K} \exp \boldsymbol{v}_i^\top \boldsymbol{t}_k + \sum_{k\neq i}^{K} \exp \boldsymbol{v}_k^\top \boldsymbol{t}_i\Big) \tag{4}$$

where N is the total number of training samples and K the number of negative samples.

7.2 Retrieval on SYMON

For the retrieval task, we create training, validation, and test sets with 4,191, 500 and 502 videos, respectively. No movies or movie franchises appear in two sets simultaneously. The videos are divided into non-overlapping clips, each consisting of two scenes and having mean duration of 4.4 seconds. YouTube videos often contain introduction and channel information at the beginning and the end, so we exclude 5% at each end of the videos.

In Table 6, we report recall at 1, 5, and 10 items (R@1, R@5, and R@10), and Median Rank (MR). As the video and the text are not exactly matched by time, given a video clip, we consider the three closest sentences as correct answers and vice versa. As we expect, the UniVL network finetuned on SYMON (UniVL-SYMON) outperforms the original UniVL weights.

7.3 Transfer to YMS

Without in-domain finetuning, we directly test the model trained on SYMON on the YMS dataset, which contains 94 YouTube movie summary videos

Model	R@1	R@5	R@10	MR
Text-to-video Retrieval				
UniVL	0.11	0.39	0.63	4818
UniVL-SyMoN	**0.73**	**2.02**	**3.07**	**1785**
Video-to-text Retrieval				
UniVL	0.03	0.11	0.19	5687
UniVL-SyMoN	**0.89**	**2.03**	**2.93**	**1843**

Table 6: Retrieval performance on SyMoN

	Clip Acc.	Sent. IoU
Original Data Split and Segmentation		
UniVL	3.7	1.5
NeuMATCH-MD (Supervised)	4.0	2.4
UniVL-SyMoN	**5.4**	**2.6**
New Data Split and Segmentation		
UniVL	4.3	2.1
UniVL-SyMoN	**6.2**	**2.4**

Table 7: Zero-shot alignment performance on YMS.

with manual annotation of fine-grained video-text alignment. To prevent test data leak, we remove any summary videos for the 94 YMS movies from the training set used in this experiment.

Evaluation. In YMS, a text segment may correspond to multiple video clips, whereas a video clip may correspond to one or zero text segment. During inference, we align every video clip to the text segment with the highest similarity, as computed by the neural network. This creates the desired many-to-one alignment. If the highest similarity falls below a threshold, tuned on the validation set, the video clip is considered as not matching anything.

Following Dogan et al. (2018), we use clip accuracy (*i.e.* the temporal proportion of correctly aligned video segments), and sentence IoU (*i.e.* the intersection-over-union metric between aligned video durations and ground-truth durations) as evaluation metrics.

Baselines. Using the network described in §7.1, we compare the original UniVL weights, UniVL finetuned on SyMoN data (UniVL-SyMoN), as well as the supervised NeuMATCH network without the sequential context (*i.e.*, the minimum distance (MD) baseline from Dogan et al. (2018)). Note that UniVL-SyMoN is trained with two video scenes as the basic unit for retrieval and NeuMATCH-MD uses more finely segmented units. As YMS contains fine-grained annotations, it is likely that this comparison puts our network at a disadvantage.

Test Data Split and Segmentation. For fair comparison with NeuMATCH-MD, we use the original test set of 15 videos and the original video segmentation. In addition, we also create a new split using 70% of the entire YMS as the test set and 30% as the validation set. In this new setting, the videos are segmented into scenes as detected in our preprocessing (§4).

Results. Table 7 shows the results. Despite the difference in segmentation and the weak supervision from SyMoN, UniVL-SyMoN outperforms the supervised NeuMATCH-MD baseline. This shows that UniVL-SyMoN learns a superior cross-modality distance metric, demonstrating the utility of the large-scale SyMoN dataset. UniVL-SyMoN also outperforms the original UniVL by 1.7% / 1.1% in the original setting and 1.9% / 0.3% in the new setting. Considering UniVL was trained on the gigantic HowTo100M dataset, we attribute the improvement to the similarity between SyMoN and YMS, which highlights the effectiveness of SyMoN in the domain of story video understanding.

8 Conclusion

In this work, we collect and process a story understanding SyMoN. We compare SyMoN with existing video-language datasets and quantitatively analyze the story coverage, the amount of mental-state descriptions, and the semantic divergence between video and text. Furthermore, we establish multimodal retrieval baselines for SyMoN and a zero-shot alignment baseline on YMS to demonstrate the effectiveness of SyMoN in story understanding. We believe SyMoN will serve as a new challenge for the research community and inspire new advances of multimodal machine learning.

9 Potential Ethical Impact

In this paper, we collect user-uploaded videos from YouTube, which are summaries of mostly western movies and TV shows in the English language. We recognize that movies and TV shows are fictional in nature, and often prioritize dramatic events over faithful representation of real-life scenarios. In addition, the videos may reflect particular bias of the creators of the movie and TV shows or the creators of the summary videos, as well as bias from particular cultures or the time periods of production.

For these reasons, we urge researchers to take caution when attempting to learn social norms from such videos. For example, events of bank robberies may be over-represented in these videos, and a machine learning model may inadvertently infer that robbing a bank is part of the social norm. In addition, the model may incorrectly learn from disproportional association of certain groups of people with certain social status, occupations, and other cultural constructs.

We further note that most relations between events are probabilistic and neither necessary nor sufficient. For example, though it is common for someone with a medical emergency to call for an ambulance, it does not always happen. We suggest researchers to similarly qualify any learned relations. The dataset is intended for fundamental research and not real-world deployment.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for resource-rich and-poor languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142.

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374.

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. 2020. Condensed movies: Story based retrieval with contextual embeddings. In *Proceedings of the Asian Conference on Computer Vision*.

Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *ICLR*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "let your characters tell their story": A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jerome Bruner. 1986. *Actual Minds, Possible Worlds*. Harvard University Press.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970.

Tommaso Caselli, Eduard Hovy, Martha Palmer, and Piek Vossen. 2021. *Computational Analysis of Storylines: Making Sense of Events*. Cambridge University Press.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III au2, and Chris Dyer. 2016. Modeling dynamic relationships between characters in literary novels. In *AAAI*.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021. Summscreen: A dataset for abstractive screenplay summarization. *arXiv Preprint 2104.07091*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pelin Dogan, Boyang Li, Leonid Sigal, and Markus H. Gross. 2018. LSTM stack-based neural multi-sequence alignment technique (neumatch). *CoRR*, abs/1803.00057.

David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th annual meeting of the* *association for computational linguistics*, pages 138–147.

Denis Emelin, Ronan Le Bras, Jena D. Hwang, Maxwell Forbes, and Yejin Choi. 2020. Moral stories: Situated reasoning about norms, intents, actions, and their consequences.

Francis Ferraro and Benjamin Van Durme. 2016. A unified bayesian model of scripts, frames and language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Varun Gangal, Steven Y. Feng, Malihe Alikhani, Teruko Mitamura, and Eduard Hovy. 2021. Nareor: The narrative reordering problem. *arXiv 2104.06669*.

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

F.G.E. Happé. 1994. An advanced test of theory of mind: Understanding of story characters' thoughts and feelings by able autistic, mentally handicapped, and normal children and adults. *Journal of Autism and Developmental Disorders*, 24:129–154.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 709–727. Springer.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Evgeny Kim and Roman Klinger. 2019. Frowning frodo, wincing leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters. In *NAACL*.

Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. 2017. Deepstory: Video story qa by deep embedded memory networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2016–2022.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA Reading Comprehension Challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. 2020. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2018. Tvqa: Localized, compositional video question answering. In *The 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Boyang Li, Beth Cardier, Tong Wang, and Florian Metze. 2018. Annotating high-level structures of short stories and personal anecdotes. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Boyang Li, Stephen Lee-Urban, George Johnston, and Mark O. Riedl. 2013. Story generation with crowdsourced plot graphs. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2018. Graph-based decoding for event sequencing and coreference resolution. In *COLING*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*.

Lara Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark Riedl. 2018. Event representations for automated story generation with deep neural nets. In *The AAAI Conference on Artificial Intelligence*, volume 32.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. Torque: A reading comprehension dataset of temporal ordering questions. In *The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172. Association for Computational Linguistics.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56, Austin, Texas. Association for Computational Linguistics.

Jessica Ouyang and Kathleen McKeown. 2015. Modeling reportable events as turning points in narrative. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2149–2158, Lisbon, Portugal. Association for Computational Linguistics.

Pinelopi Papalampidi, Frank Keller, Lea Frermann, and Mirella Lapata. 2020. Screenplay summarization using latent narrative structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1920–1933, Online. Association for Computational Linguistics.

Janette Pelletier and Ruth Beatty. 2015. Children's understanding of aesop's fables: relations to reading comprehension and theory of mind. *Frontiers in Psychology*, 6:1448.

Karl Pichotta and Raymond J. Mooney. 2016. Using sentence-level LSTM language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–289, Berlin, Germany. Association for Computational Linguistics.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. Event2Mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*.

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

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

Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ram Nevatia, and Aniruddha Kembhavi. 2021. Visual semantic role labeling for video understanding. *CoRR*, abs/2104.00990.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *The AAAI Conference on Artificial Intelligence*.

Advaith Siddharthan, Nicolas Cherbuin, Paul J Eslinger, Kasia Kozlowska, Nora A Murphy, and Leroy Lowe. 2018. Wordnet-feelings: a linguistic categorisation of human feelings. *arXiv preprint arXiv:1811.02435*.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer.

Tomáš Souček and Jakub Lokoč. 2020. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*.

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

Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM): the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. 2018. Moviegraphs: Towards understanding human-centric situations from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2021. Joint constrained learning for event-event relation extraction. In *EMNLP*.

Chao-Yuan Wu and Philipp Krahenbuhl. 2021. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321.

Zhengzhe Yang and Jinho D. Choi. 2019. FriendsQA: Open-domain question answering on TV show transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 188–197, Stockholm, Sweden. Association for Computational Linguistics.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

A Story Coverage

Dynamic Time Warping We present the DTW problem formulation: given the WikiPlots sequence of sentences $A = (a_1, \ldots, a_N)$ and the video narration sentences $B = (b_1, \ldots, b_M)$, we seek the best set of correspondence $\{(a_i, b_{g(i)})\}_{i=1}^{N}$, where the function $g(i) \in \{\epsilon, 1, \ldots, M\}$ returns the index in sequence B that matches sentence a_i in A. Setting $g(i) = \epsilon$ indicates that a_i is not matched with any sentence in B.

The DTW algorithm can be understood as finding the shortest path in a graph, where each node (i, j) in the graph represents matching sentence a_i and sentence b_j. The graph contains dummy nodes $(0, 0)$ and $(N + 1, M + 1)$. From node (i, j), we can transit to node $(i + 1, j + 1)$, which would match a_{i+1} with b_{j+1} and incur cost $c(i+1, j+1)$.

$$c(i + 1, j + 1) = 1 - P(a_{i+1} \Leftrightarrow b_{j+1}). \quad (5)$$

Here $P(a_{i+1} \Leftrightarrow b_{j+1})$ denotes the probability that sentences a_{i+1} and b_{j+1} are equivalent, as determined by the Natural Language Inference classifier.

Similarly, we can transit from (i, j) to $(i + 1, j)$, which would match a_{i+1} with b_j and incur cost $c(i+1, j)$. The transition from (i, j) to $(i, j+1)$ is symmetric. Additionally, we can transit from (i, j) to $(i, j + 1, \epsilon)$, which prevents b_{j+1} from matching anything. From $(i, j + 1, \epsilon)$, we may transit to $(i, j + 2, \epsilon)$, $(i, j + 2)$, or $(i + 1, j + 2)$. The costs of ignoring a sentence in A and B are δ_A and δ_B respectively. With this setup, the best correspondence can be found as the path from $(0, 0)$ to $(N + 1, M + 1)$ with minimum cost. We find optimal δ_A and δ_B using manually labeled sentence correspondence.

Annotation instructions Fig. 4 shows the instructions we give to our annotator. Here column A is the WikiPlot summary and column B is the summary from SYMON or CMD or LSMDC.

B Video Temporal Ordering

Hyperparameters. We sample each video segment at 16 frames per second (FPS) and extract features with S3D (Xie et al., 2018) pretrained on HowTo100M. Each video video segment last exactly 8 seconds. We extract S3D features every second (i.e. from 16 frames), yielding 8 1024-dimensional video features for each video segment. For video features extraction we use frame size of 112×112.

We extract the text between the start of the first video segment and the end of the second video segment. To ensure completeness, the text is extended to the nearest sentence boundaries. The maximal

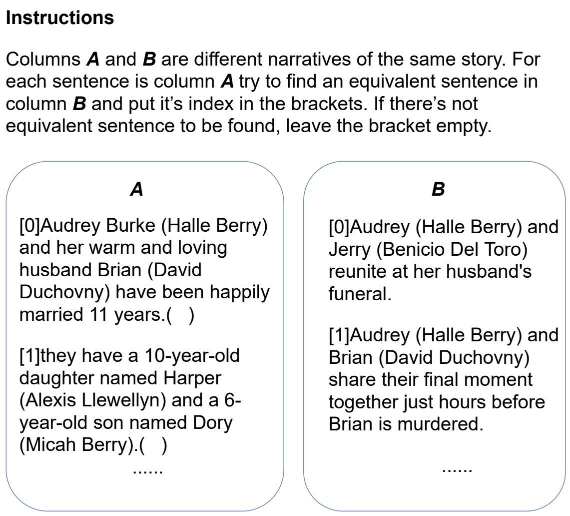

Figure 4: Annotation Instruction

number of text tokens is 128. For longer texts, we remove extra tokens from the start and end of the text. For shorter texts, we add zero padding to the end.

The text encoder, video encoder, and cross encoder consist of 12, 6, 2 Transformer layers, respectively. The models are trained for 30 epochs with learning rate warm-up in the first 6 epochs. Hyperparameters are tuned on the validation set. The text-aware model is trained with a batch size of 128 and learn rate of $5e-6$ and the visual-only model is trained with a batch size of 256 and initial learning rate of $1e-5$. We apply cosine learning rate decay and the Adam optimizer to all models.

The model contains 217,185,539 parameters and is trained for 2.7 hours on 4 Nvidia 3090 GPUs. The results reported in the main paper are on a single run.

Calculating overlap between text description and object/action class We first tokenize the text description and use part-of-speech tagging to identify nouns and verbs in the text description (Bird et al., 2009). For matching with object and action detection, we retain the nouns and verb from text description, respectively. We also lemmatize the retained words to remove variations and remove common nouns and verbs ("men", "women", "person", and "clothing" for nouns and "is", "go", "to", "get", "have", "look", "walk", "play", and "take" for verbs). For object detection we retain the top 10 class predictions for each clip. For action detection we divide the clip into scenes and retain the top 3 class prediction for each scene. Finally, we calculate the number of time the retained nouns or verbs appear in the detected object or action class names.

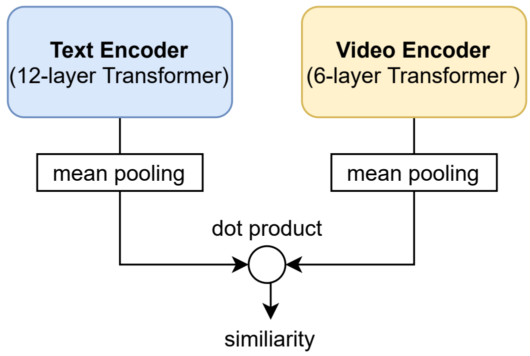

Figure 5: Retrieval Model

C Retrieval

Hyperparameters. For video feature extraction we sample the video at 16 FPS and use S3D pretrained on HowTo100m to extract one 1024 dimensional feature every 16 frames. The frame size is 112×112. For each clip we extract 4 features, if the clip is shorter than 4 seconds zero padding is add and if the clip is longer than 4 seconds we only use the first 4 second. Likewise, we take 64 text tokens for each clip. Text is extracted from between the start ans end of the video clip and extended to the nearest sentence boundaries. The video and text encoders consist of 12 and 6 transformer layers respectively, and are initialized from UniVL pretrained on HowTo100m. The outputs are then averaged into two 768 dimensional embeddings for video and text. The similarity between a video, text pair is calculated as the dot product of the video and text embeddings. The model is finetuned on SYMON with an initial learning rate of $5e-5$ and cosine learning rate decay. We use a batchsize of 1024 and train for 20 epoches, the first epoch is warm up. SGD with momentum of 0.9 is used for optimization and s weight decay term of 0.5 is added for regularization.

The model contains 153,784,064 parameters and is trained for 4 hours on 4 Nvidia 3090 GPUs. The results reported in the main paper are on a single run.

D Implementation and Licensing Details

For the subtitle masking in §4 we used EasyOCR (Baek et al., 2019) for image character recognition. The model we use is the english_g2 model from https://www.jaided.ai/easyocr/modelhub/. EasyOCR (Baek et al., 2019) is licensed under the Apache License, Version 2.0.

For the punctuation restoration in §4 we used the network from Alam et al. (2020). The model we use is given here https://drive.google.com/file/d/17BPcnHVhpQlsOTC8LEayIFFJ7WkL00cr/view. The network and model are released under the MIT license.

For scene segmentation in §4 we used TransNet-V2 (Souček and Lokoč, 2020) to identify scene boundaries, the network weight are from think link https://github.com/soCzech/TransNetV2/tree/master/inference/transnetv2-weights. For every frame, the network predict the probability of a scene change occurring immediately after the frame, if the probability is larger than a threshold of 50%, we deem a scene change had occured. TransNet-v2 (Souček and Lokoč, 2020) is released under the MIT license.

For entailment prediction in §5.1 we use AdversarialNLI (Nie et al., 2020), specifically the 'roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli' model. AdversarialNLI (Nie et al., 2020) is released under the MIT licence. For this section we use WikiPlot summaries from https://github.com/markriedl/WikiPlots as ground truth movie summaries. The release does not include a license. Additionally, we compare our dataset to the CMD (Bain et al., 2020) dataset and LSMDC (Rohrbach et al., 2017) dataset, both are released under the Creative Commons Attribution 4.0 International License.

For the mental-state description experiment in §5.2, we collect emotion related words from WordNet-feelings (Siddharthan et al., 2018) dataset. The release does not include a license. We collect intention, motivation related words from the top 200 nearest neighbors on Fast-text (Girshick, 2015) word embedding, which is acquired from https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md. fast-Text (Girshick, 2015) is released under the MIT licence. The word embedding is release under the Creative Commons Attribution-Share-Alike License 3.0. The ActivityNet dataset (Krishna et al., 2017) is licensed under the MIT license.

For the video sequencing experiment in §6, we use UniVL (Luo et al., 2020) pretrained on HowTo100m (Miech et al., 2019). The model weights are initialized from https://github.com/microsoft/UniVL/releases/download/v0/univl.pretrained.bin. UniVL (Luo et al., 2020) and HowTo100m (Miech et al., 2019) are licensed under MIT and Apache License 2.0 respectively.

For object recognition in §6, we use Faster-RCNN (Girshick, 2015) trained on Open Images V4 (Kuznetsova et al., 2020) to detect objects from video frames, and 3D-ResNet (Hara et al., 2018) trained on Kinetics-700 (Kay et al., 2017) to detect actions. Faster-RCNN (Girshick, 2015) and 3D-ResNet (Hara et al., 2018) are licensed under the MIT license. Open Images V4 is released under Apache License 2.0. Kinetics-700 is licensed under the Creative Commons Attribution 4.0 International License. The text descriptions are processed with the nltk package (Bird et al., 2009), licensed under Apache License 2.0.

For the multimodal retrieval task in §7, we use the YMS dataset (Dogan et al., 2018) from https://github.com/RubbyJ/Data-efficient-Alignment.