Video Discourse Parsing and Its Application to Multimodal Summarization: A Dataset and Baseline Approaches

Tsutomu Hirao[1]*, Naoki Kobayashi[2]*, Hidetaka Kamigaito[2],
Manabu Okumura[2], Akisato Kimura[1]
[1]NTT Communication Science Laboratories, NTT Corporation
[2]Tokyo Institute of Technology,
{tsutomu.hirao, akisato.kimura}@ntt.com
{kobayasi@lr., kamigaito@lr., oku@}pi.titech.ac.jp

Abstract

This paper tackles a task: discourse parsing for videos, inspired by text discourse parsing based on Rhetorical Structure Theory (RST). The task aims to construct an RST tree for a video to represent its storyline and illustrate the event relationships. We first construct a benchmark dataset by identifying events with their time spans, providing corresponding captions, and constructing RST trees with events as leaves. We then evaluate baseline approaches to video RST parsing: the 'parsing after captioning' framework and parsing via visual features. The results show that a parser using gold captions performed the best, while parsers relying on generated captions performed the worst; a parser using visual features provided intermediate performance. However, we observed that parsing via visual features could be improved by pre-training it with video captioning designed to produce a coherent video story. Furthermore, we demonstrated that RST trees obtained from videos contribute to multimodal summarization consisting of keyframes with texts.

1 Introduction

Videos often consist of several parts, including an introduction, development, turn, and conclusion, which together form a coherent plot to effectively convey a story.[1] Even shorter videos, such as consumer-generated videos lasting only a few minutes, possess similar story structures. The primitive units in the video that make up the story structure are called events,[2] which are used to develop and advance the story (Li et al., 2020a).

The recent success of discourse parsing, particularly Rhetorical Structure Theory (RST) style discourse parsing, e.g., Kobayashi et al. (2022), inspires us to introduce it to help us understand video stories. Consider the significance of discourse parsing for video summarization tasks. For instance, the video in Fig. 1 captures the high jump competition at Summer Universiade held in Taipei. It focuses on three competitors, displaying their attempts at different event spans. Then, it features the winner of the competition, which is the third competitor. The first competitor's event span is shown between seconds 9.6 and 21.5. The second competitor's event span is between seconds 21.6 and 27.9, and the third competitor's event span is between seconds 32.1 and 50.7. It is worth noting that the second competitor also joined the lap of honor. Consequently, omitting the third competitor's span in a summary may mislead viewers into believing that the second competitor is the winner (see Appendix A). This highlights the requirement that a summary must maintain the structure of the story in the original video. That is, we need to understand how events are related to each other in order to form a coherent storyline. Discourse parsing is a promising solution to this issue and can be helpful in many tasks that depend on video comprehension, such as video summarization (Gygli et al., 2014), video storytelling (Li et al., 2020b), video QA (Zhong et al., 2022), and other related tasks.

This paper focuses on discourse parsing for videos (hereinafter, video RST parsing), drawing inspiration from RST-style text discourse parsing approaches. Since this is still a relatively new task, we begin by building a benchmark dataset. We constructed a reliable benchmark dataset for the task, Video Discourse TreeBank (VDTB), comprising 1,100 videos obtained from YouTube. This dataset includes annotations of events with time spans and

*Equal contribution.

[1] Note that the video story structures addressed in this paper are not confined to such patterns.

[2] An event indicates a video span (segment), a kind of logical story unit (Hanjalic et al., 1999). Recently, there has been reasonable improvement in segmentation performance, as demonstrated by Ji et al. (2022).

Findings of the Association for Computational Linguistics: EACL 2024, pages 9943–9958
November 12-16, 2024 ©2024 Association for Computational Linguistics

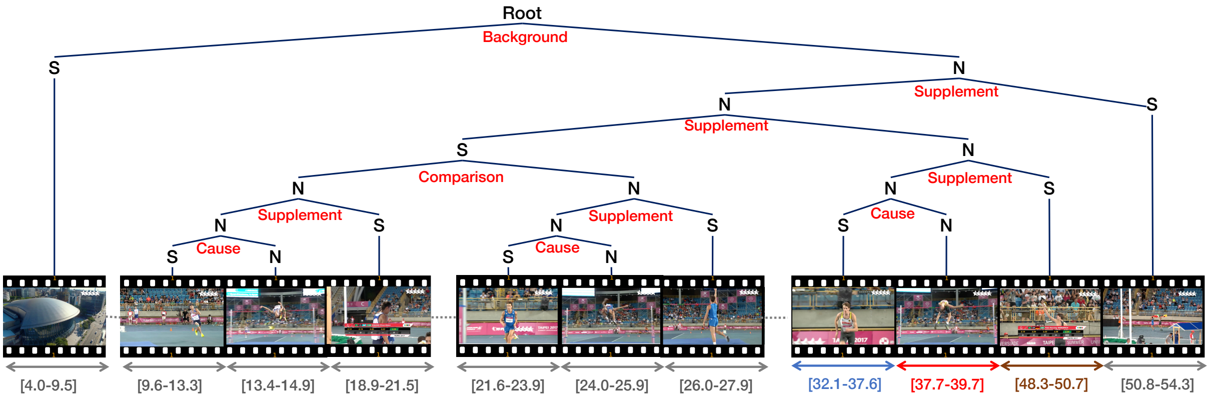


Figure 1: Example of annotations for a video

captions.[3] Then, we compare 'parsing after captioning' using a given text for a video and 'direct video parsing' based on visual features to address the research question of which approach is better suited for video RST parsing. The results show that while the former approach is comparable to human performance when using gold captions, its performance is worse when using generated captions. The latter provided an intermediary performance between the above two caption-based parsers. We further observed significant improvement in performance by pre-training this parser with video captioning designed to generate the story of the video. These findings highlight the potential benefits of incorporating both visual and textual features in video RST parsing. Furthermore, we provide annotations for multimodal summarization based on keyframes and text for the VDTB test set. Then, we highlight that video RST trees enhance the performance of multimodal summarization. We will release our dataset (annotations for videos) at https://github.com/titech-nlp/VideoParsing_EMNLP24

2 Preliminary

According to RST, a text is represented as a constituent tree whose leaves are an Elementary Discourse Unit (EDU), i.e., a clause-like unit, and whose intermediate nodes represent the nuclearity status (nucleus (N) or satellite (S)) of a text span consisting of EDU(s). Since an RST tree can be represented as a binary tree, a mono-nuclear relation (S-N or N-S), such as 'Elaboration' and 'Attribution,' or a multi-nuclear relation (N-N), such as 'List' and 'Same-Unit,' is given as an edge label between two sibling intermediate nodes. The set of rhetorical relations depends on the text domain.

Let us illustrate the annotation for discourse structures in a video, inspired by RST, by again using Fig. 1. Time spans for events can be identified in the video, including the introduction, attempts by the three competitors, and the winner of the competition. The relations between the events are represented by a constituent tree according to the nature of RST. The event with the blue-colored time span, the approach of the third competitor from second 32.1 to 37.6, modifies the event with the red-colored time span, his jump from second 37.7 to 39.7, using the rhetorical relation 'Cause.' The nuclearity statuses of the events of the blue- and red-colored time spans are satellite and nucleus, respectively. Then, the event with the brown-colored time span, the third competitor's best record from second 48.3 to 50.7, modifies the event consisting of the blue- and red-colored time spans using the rhetorical relation 'Supplement.' The nuclearity status of the event of the brown-colored time span is satellite, and that of the event consisting of the blue- and red- colored time spans is nucleus.

3 Related Work

Action parsing is a popular method for identifying local relationships between atomic actions in a video. This is achieved by identifying temporal action segments and determining their corresponding action labels (Richard et al., 2017a,b). Most studies on action parsing focus on understanding the structure within events rather than the overall storyline for a video (Shao et al., 2020). For example, a complex action, such as 'hammer throw', may be broken down into smaller atomic actions, such as 'swing,' 'rotate the body,' and 'throw.' Some stud-

[3] Captions serve as auxiliary information to explore the upper bound performance of parsing. See Section 5.3 for more details.

ies, however, delve deeper into the relationships among atomic actions. Kuehne et al. (2014) represented an event as a graph with nodes as atomic actions and edges as the relations between them. Luo et al. (2021) introduced hypergraphs to represent the relationships among events. Such structures are useful for predicting intent (Qi et al., 2018), future actions (Pei et al., 2011), and answering questions (Tu et al., 2013). These studies are similar to ours in terms of analyzing the structure for videos; however, our study aims to represent the overall coherency of the storyline in a video, that involves long-distance relationships between events.

Watanabe et al. (2000) proposed a method for RST parsing of news videos with the aim to capture the overall storyline in the video. Their approach assumes shots as basic discourse units and constructs RST structures based on them. Each discourse unit in a video corresponds to a transcript, which is utilized to parse the video's RST tree using manually designed rules. Importantly, this method solely relies on text and does not incorporate any features from the videos themselves.

In a recent preliminary study, Akula and Song-Chun (2020) proposed a video discourse parsing approach based on a 'parsing after captioning' framework. This process involves generating a few captions for a given video, predicting an RST tree using these captions, and subsequently aligning the video frames and the captions to construct the RST tree. It's important to note that the primitive discourse units in their approach are video frames. That is, their approach presents a limitation in handling longer videos because the leaf nodes of their RST trees correspond to video frames rather than events. As a result, the dataset mainly includes very brief videos, with an average duration of mere 19 seconds, and it is accompanied by only a few captions. Additionally, there are no annotations for event time spans. Accordingly, due to the limited number of captions, constructing a meaningful RST tree seems impractical.

Although both studies tackled the issue of video discourse parsing, their focus leaned towards using text discourse parsing instead of purely video discourse parsing. Nevertheless, they have major limitations, such as the general lack of high-quality transcripts or captions that can be used for text discourse parsing.

4 Constructing a Dataset

We construct Video Discourse TreeBank (VDTB), a dataset for video discourse parsing, as there are no existing datasets available for this purpose. From YouTube, we selected 1,100 videosshowing human activities of less than a few minutes. We manually searched for channels on YouTube that featured news, sports, cooking, DIY, and other activity-oriented topics. We carefully selected videos with engaging stories that were free of sensitive or potentially harmful content. The domains of the videos included news (24%), activities (13%), sports (13%), instructions (15%), homemade videos (16%), and misc (19%). Next, we asked two annotators, with backgrounds in natural language processing, to identify events. The annotators then generated a concise caption, preferably using simple sentences with generally one subject and one verb, for each event to form a story of the entire video. This guidance aimed to avoid detecting lengthy events consisting of several events. After that, they built an RST tree for each video based on the events by following the instructions for constructing RST Discourse Treebank (RST-DT) (Carlson et al., 2001).

Note that 50 videos in VDTB, separated as the test set, have two different annotations by two different annotators to verify the inter-annotator agreement. That is, each of the 50 videos has two different sets of events, captions, and RST trees made by the different annotators. The detailed procedure is shown in Appendix B.

Since the domains of VDTB and RST-DT are different from each other, we refined the rhetorical relations used in RST-DT for our purposes by introducing new rhetorical relations and omitting other relations.[4] We show our nine rhetorical relations assigned between two intermediate siblings in Appendix C. When dealing with new video domains, we may need to adjust rhetorical relations by introducing new ones or modifying existing ones, similar to the process in text RST parsing.

5 Inter-annotator Agreement in Event Detection and Captioning

VDTB provides annotations for both event spans and their corresponding captions. These annotations are similar to those in ActivityNet Captions

[4] Since the leaves of our RST tree are events written with a single caption, we omitted all intra-sentential rhetorical relations.

	ANC		VDTB	
	A	B	A	B
Duration (sec.)	118.2		105.4	
Num. of Events	3.49	3.56	10.4	9.74
Len. of Events (sec.)	40.2	37.7	8.73	9.71
Words/sent.	14.2	12.7	9.78	10.8

Table 1: Statistics of datasets

(Krishna et al., 2017), which is used for dense video captioning (DVC), with one key distinction: VDTB does not allow events with temporal overlaps. This non-overlapping structure, also employed in other datasets such as YouCook2 (Zhou et al., 2018a), is particularly well-suited for domains where events typically occur sequentially

To validate the annotation quality of VDTB, we compared the inter-annotator agreement for event detection and caption generation in VDTB with those in ActivityNet Captions, which also has annotations for them.[5]

5.1 Properties of Datasets

The statistics of VDTB and ActivityNet Captions are summarized in Table 1. While the table shows no significant difference in the average duration of videos between the two datasets, there are significant differences in the average number and length of events. While ActivityNet Captions has 3.6 events per video on average, VDTB has approximately 10 events. Furthermore, there is a significant difference in the length of events. While the average length of events in ActivityNet Captions is around 40 seconds, that of VDTB is approximately 10 seconds. Since there is no significant difference in the average number of words per caption between the two datasets, the above properties imply a substantial difference in the granularity of events identified by the annotators.

5.2 Event Detection

To validate the inter-annotator agreement of event detection, we first tried to find one-to-one matching between event time spans from different annotators. We applied SODA (Fujita et al., 2020), which is an evaluation metric for DVC, to find the one-to-one matching of events that maximizes the sum of tIoU (temporal Intersection of Union) between the event time spans. Here, tIoU is defined as

	ANC		VDTB	
	A → B	B → A	A → B	B → A
tIoU	.413	.422	.621	.666
Match	.759	.776	.822	.881

Table 2: Average tIoU and the ratio of matching between event time spans. A→B indicates the score when regarding the annotations by annotator A as the reference and those by B as the hypothesis, with B←A being vice versa.

$$\mathrm{tIoU}(g,p) = \max\left(0, \frac{\min(\mathrm{e}(g),\mathrm{e}(p)) - \max(\mathrm{s}(g),\mathrm{s}(p))}{\max(\mathrm{e}(g),\mathrm{e}(p)) - \min(\mathrm{s}(g),\mathrm{s}(p))}\right), \quad (1)$$

where g and p are events, and functions s() and e() return the start and end times of the events, respectively. The one-to-one matching can be found by filling the DP table as follows:
Initialization: Recurrence: $(1 \leq i \leq |\mathcal{P}|, 1 \leq j \leq |\mathcal{G}|)$

$$S[i][j] = \max \begin{cases} S[i-1][j], \\ S[i-1][j-1] + C_{i,j}, \\ S[i][j-1], \end{cases} \quad (2)$$

where $\mathcal{G}, \mathcal{P}$ is a set of events and $C_{i,j}$ is tIoU between the i-th event in $\mathcal{G}$ and the j-th event in $\mathcal{P}$.

Table 2 shows the micro-averaged tIoU between events and the ratio of events for which we could find a match. From the table, we can see that the averaged tIoU obtained from VDTB was higher, at a significant level, than that obtained from ActivityNet Captions. Furthermore, VDTB was superior to ActivityNet Captions in terms of the ratio of matched events. These results suggest that the annotation for event detection in VDTB is more consistent than that in ActivityNet Captions.

5.3 Caption Generation

Our objective is to create RST-style discourse trees whose leaves are events from a given video. It is not a requirement for each event to have a caption; however, we provided captions for a later comparison of different parsing methods: 'parsing after captioning' and 'direct parsing with visual features.' In addition, creating captions for events may ease annotators in constructing RST trees.

We evaluated each annotator's captions with automatic evaluation metrics by employing captions generated by the other annotator as the reference. Here, we employed ActivityNet Score (Krishna et al., 2017), a de facto standard evaluation metric, and SODA (Fujita et al., 2020), a recently proposed story-aware evaluation metric.

[5] The validation set of ActivityNet Captions, consisting of 4,885 videos, has two different annotations for events and their corresponding captions.

	ANC		VDTB	
	A→B	B→A	A→B	B→A
ANetSc	6.00	6.51	18.6	17.8
SODA	5.83	5.23	15.8	16.3

Table 3: ActivityNet Score and SODA using the captions by one annotator as the reference and those by the other annotator as the hypothesis

	RST-DT		VDTB	
	A	B	A	B
Num. of sent.	22.8		10.4	9.74
Words/sent.	21.3		9.78	10.8
N-S (%)	57.4	59.7	35.2	33.6
S-N (%)	16.6	16.4	47.7	53.0
N-N (%)	26.0	23.9	17.1	13.4

Table 4: Statistics of the parts of two datasets, VDTB and RST-DT, each having double annotations

Table 3 presents the results. The scores for VDTB were significantly higher than those of ActivityNet Captions, with differences of approximately 10 points for both evaluation metrics. These results suggest that our annotations, performed by two annotators, are more consistent than those in ActivityNet Captions in terms of both event detection and caption generation.

However, it is important to note that high inter-annotator agreement may reflect the simplicity of video interpretation. Our current study primarily focuses on establishing how well models can parse easily interpretable videos as an initial step in this research direction.

6 Inter-annotator Agreement in RST Tree Construction

In this section, we compare VDTB and RST-DT for their RST tree qualities.

6.1 Properties of Datasets

As a part of RST-DT, 53 out of 385 documents have two RST trees[6] annotated by different annotators. In Table 4, we show the following statistics for RST-DT and VDTB: average number of sentences per document, average number of words per sentence, and distribution of nuclearity labels for sibling nodes.

From Table 4, the average number of sentences per document and that of words per sentence in RST-DT are twice as large as those in VDTB.

	Span	Nuc.	Rel.	Full
RST-DT	58.6	43.2	30.2	28.8
VDTB(A):A↔B	68.0	55.7	46.8	46.6
VDTB(B):A↔B	67.5	52.4	47.3	46.7

Table 5: Results of Standard-Parseval (Morey et al., 2017) for the annotations by one annotator as the reference and those by the other annotator as the hypothesis. VDTB (A) and VDTB (B) indicate RST trees based on events identified by annotators A and B, respectively.

The two datasets' distributions of nuclearity labels for sibling nodes are also different from each other. While the majority in RST-DT is N-S, that in VDTB is S-N. We believe this difference comes from their writing styles. Since the source text of RST-DT is a newspaper article, important information of the nucleus comes first, and then the details of the satellite are described. On the other hand, since captions in VDTB describe events along a timeline, important information of the nucleus tends to be presented later. Furthermore, frequent rhetorical relations in VDTB are also different from those in RST-DT. We show the detailed distribution of the rhetorical relations in VDTB and RST-DT in Appendix D.

6.2 RST Tree Construction

We evaluated one annotator's RST tree by employing that of the other annotator as the reference tree. Since the two different annotators each identified two separate events for a video, four different RST trees were constructed for a single video (see Appendix B). We used micro-averaged F1 scores of unlabeled spans (Span), those of nuclearity-labeled spans (Nuc.), those of rhetorical relation-labeled spans (Rel.), and those of fully-labeled spans (Full) based on Standard-Parseval (Morey et al., 2017).

Table 5 shows the results. From the table, we can see that VDTB obtained better scores than RST-DT. While the scores degraded in the order of Span, Nuc., Rel., and Full in both datasets, the degradation in VDTB was less sensitive than in RST-DT. These results demonstrate that the annotation in VDTB is more consistent than that in RST-DT. However, we do not believe these results suggest that RST-DT has low consistency between annotators. Rather, we think the results seem natural because documents in RST-DT are longer and RST-DT requires more rhetorical relations for annotation. In fact, the number of relations in RST-DT is approximately twice as large as in VDTB.

[6] In this paper, we assume that the RST trees of RST-DT are transformed so that their leaves are sentences in order to make a fair comparison with the RST trees in VDTB.

7 Baseline Parsers

We developed two parsers, the ‘parsing after captioning’ framework and parsing via visual features, with the same architecture, except for the feature extraction module.

7.1 Parsing Model

As our base RST parsing model, we employed Kobayashi’s transition-based bottom-up parser (Kobayashi et al., 2022), [7] which is one of the SOTA parsers for text RST parsing. We chose this model due to its simplicity and its suitability as a baseline for video discourse parsing.

Fig. 2 shows the architecture of our parser. In the figure, a stack stores subtrees, i.e., event spans that have already been parsed, and a queue contains incoming events. The parser builds an RST tree in a bottom-up manner by merging two adjacent event spans while choosing one of the following actions: **Shift**: Pop the first event off the queue and push it onto the stack. **Reduce**: Pop two subtrees from the stack and merge them into a new subtree, then push it onto the stack. Note that the nuclearity status and relation labels are independently predicted by different classifiers. FFN_{act}, FFN_{nuc}, and FFN_{rel} are feed-forward neural networks for predicting an action, nuclearity, and relation labels, respectively. FFN_{act} solves a binary classification problem (Shift or Reduce), FFN_{nuc} solves a three-class classification problem (N-S, N-N, or S-N), and FFN_{rel} solves a nine-class classification problem (nine is the number of rhetorical relations): $s_*{=}\text{FFN}_*(\text{Concat}(\mathbf{u}_{\text{s}_0}, \mathbf{u}_{\text{s}_1}, \mathbf{u}_{\text{q}_0}))$, where the function “Concat” concatenates the vectors received as the arguments. $\mathbf{u}_{\text{s}_0}$ is the vector representation of an event span stored in the first position of the stack, $\mathbf{u}_{\text{s}_1}$ is that in the second position, and $\mathbf{u}_{\text{q}_0}$ is that in the first position of the queue. The weight for each FFN and the encoder used to obtain vectors for event spans are trained by optimizing the cross-entropy loss of s_{act}, s_{nuc}, and s_{rel}.

7.2 Vector Representations for Event Spans

We need vector representations for event spans, $\mathbf{u}_{\text{s}_0}$, $\mathbf{u}_{\text{s}_1}$, and $\mathbf{u}_{\text{q}_0}$, to parse videos. Two different encoders are used to obtain vector $\mathbf{u}$. The first exploits a pre-trained language model to encode a sequence of tokens for ‘parsing after captioning,’ while the second exploits a pre-trained video encoder to encode a sequence of video frames for direct video parsing.

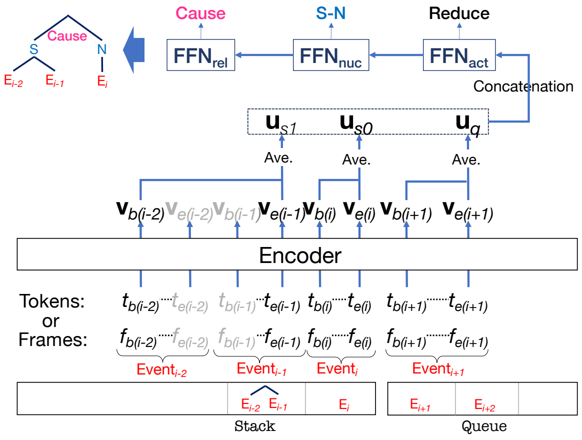

Cause
S
N
E_{i-2}
E_{i-1}
E_i
Cause
S-N
Reduce
FFN_{rel}
FFN_{nuc}
FFN_{act}
Concatenation
$\mathbf{u}_{s1}$
$\mathbf{u}_{s0}$
$\mathbf{u}_q$
Ave.
Ave.
Ave.
$\mathbf{v}_{b(i-2)}$ $\mathbf{v}_{e(i-2)}$ $\mathbf{v}_{b(i-1)}$ $\mathbf{v}_{e(i-1)}$ $\mathbf{v}_{b(i)}$ $\mathbf{v}_{e(i)}$ $\mathbf{v}_{b(i+1)}$ $\mathbf{v}_{e(i+1)}$
Encoder
Tokens:
or
Frames:
$t_{b(i-2)}$ $t_{e(i-2)}$ $t_{b(i-1)}$ $t_{e(i-1)}$ $t_{b(i)}$ $t_{e(i)}$ $t_{b(i+1)}$ $t_{e(i+1)}$
$f_{b(i-2)}$ $f_{e(i-2)}$ $f_{b(i-1)}$ $f_{e(i-1)}$ $f_{b(i)}$ $f_{e(i)}$ $f_{b(i+1)}$ $f_{e(i+1)}$
Event_{i-2}
Event_{i-1}
Event_i
Event_{i+1}
E_{i-2} E_{i-1}
E_i
E_{i+1}
E_{i+2}
Stack
Queue

Figure 2: Bottom-up parsing model

For ‘parsing after captioning,’ we transform captions for a video into a sequence of subwords, $\{t_1, t_2, \ldots, t_n\}$. Then, we obtain vector representations for the subwords as a sequence $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ by using a language model, DeBERTa v3 (He et al., 2021). Next, a vector for an event span $\mathbf{u}_{i:j}$, consisting of the i-th event to the j-th event, is obtained by averaging the vectors for two edge subwords, i.e., $\mathbf{u}_{i:j} = (\mathbf{v}_{\text{b}(i)} + \mathbf{v}_{\text{e}(j)})/2$, where $\text{b}(i)$ returns the index of the leftmost subword in the i-th event and $\text{e}(j)$ returns that of the rightmost subword in the j-th event.

For direct video parsing, we use Temporally-Sensitive Pre-training (TSP) (Alwassel et al., 2020) to directly transform event spans into vectors.[8] TSP was trained with two tasks: classifying an action type for a clip and classifying whether a clip is inside or outside an action. TSP has been widely applied in temporal action localization and dense video captioning. To obtain vector representations for frames, we applied TSP to an entire video and obtained vector $\mathbf{v}$ using a two-layer transformer (Vaswani et al., 2017) encoder. A vector for an event span, $\mathbf{u}_{i:j}$, was obtained by averaging the vectors for two edge frames, as done for textual features.[9] More implementation details are given

[7] https://github.com/nttcslab-nlp/RSTParser_EMNLP22

[8] While our experiments began by using gold event spans, using automatic event identification is crucial for future work.

[9] We might consider employing a combination of both visual and textual features. For example, a simple concatenation of vectors obtained by the above two models or their fusion by the gate mechanism might be effective; however, as described in Section 8.3, since parsing performances with automatically generated captions are rather low, such a combination would not work well. Therefore, we do not consider combinations of visual and textual features in this paper.

in Appendix E.

8 Experiments

8.1 Dataset

We used VDTB for training and evaluating the baseline parsers. We split VDTB into 1,000, 50, and 50 videos for training, validation, and test sets, respectively. Since the 50 videos in the test set have two different events, captions, and RST trees by the different annotators, we regarded the two annotations as different data and assumed that the test set consists of 100 data.[10] We also offer annotations for multimodal summarization for the VDTB test set. Two annotators generated summaries using the following process: first, an annotator creates a text summary for a video and then selects video frames (keyframes) that illustrate the content of the text summary. The average number of keyframes in a summary is 5.52, with an average of 52.5 words.

8.2 Evaluation Metrics

As the first step in video RST parsing, we used ground truth events in the evaluation experiments. While it would be more realistic to include event detection in the evaluation process, it poses a significant challenge to evaluate RST trees with automatically detected events. We evaluated an RST tree from the baseline parsers by adopting a manually created RST tree as the reference tree. Similar to the evaluation of the inter-annotator agreement in Section 5, we used micro-averaged F1 scores of unlabeled spans (Span), those of nuclearity-labeled spans (Nuc.), those of rhetorical relation-labeled spans (Rel.), and those of fully-labeled spans (Full) based on Standard-Parseval.

We assessed multimodal summaries using the Event Agreement rate and F1 scores of ROUGE-1, -2, and -L.[11] The Event Agreement rate indicates the percentage of matched events between the gold and predicted summaries. Since the video frames are timestamped, we aligned the two events based on the timestamps. In other words, we regarded a video frame as an event that contains it, and then evaluated the agreement of the events.

	Span	**Nuc.**	**Rel.**	**Full**
Visual Features	38.3	25.4	19.8	18.8
Parsing after captioning				
w/ Gold Caption	64.7	49.0	43.6	42.3
w/ SwinBERT	37.7	14.2	11.2	9.42
w/ Video Captioning	35.8	15.1	11.2	10.3
w/ Video Description	32.6	13.0	10.2	9.41
Left Branching	19.3	7.48	2.86	2.86
Right Branching	33.1	19.0	7.26	7.26
Human	67.8	54.1	47.1	46.6

Table 6: Evaluation results of video RST parsing on VDTB test set. Scores are the average of five trials with different seeds.

8.3 Experimental Results on Parsing

Table 6 shows our results. We include the results for the simple baseline methods of Left and Right Branching. The former indicates left-heavy binarized RST trees with the most frequent labels, i.e., S-N and 'Preparation' for nuclearity status and rhetorical relation labels, respectively. The latter indicates right-heavy binarized RST trees with the same most frequent labels. Human indicates manual parsing by humans. In Section 5, we evaluated RST trees from one annotator by regarding the RST trees from the other annotator as the reference for determining inter-annotator agreement. Human is the average of these two agreement scores in Table 5. We evaluated the 'parsing after captioning' framework using both gold captions and captions generated by SOTA captioning models, SwinBERT (Lin et al., 2022), Video Captioning (Zhou et al., 2018b), and Video Description (Zhu et al., 2022). The details of the automatic captioning methods and the performance evaluations can be found in the Appendix F.

From the table, we can see that Gold Caption completely outperformed Visual Features. Its performance is comparable to Human. We believe that the results rely on the quality of vector representations for event spans. Specifically, the vectors from visual features are less suitable than those from gold captions for classifying specific actions into either shift or reduce, since the vectors from visual features are similar to each other.

On the other hand, when employing automatically generated, we found significant performance degradation. All methods with automatically generated captions were outperformed by Visual Features. In particular, the differences in Nuc. Rel., and Full were remarkable. As mentioned above, since the event spans are similar in terms of visual

[10] We employed two self-contained annotations by each annotator, where an annotator annotated an RST tree for her/his own events and captions.

[11] The option used for ROUGE-1 is `-s -m -n 1 -A`, and the one used for ROUGE-2 and -L is `-m -n 2 -A`.

features, the generated captions are also similar to each other, which indicates that the generated captions are also not suitable for classifying the parsing actions; while the average BLEU score between two adjacent captions in the gold captions is around 9, that in the generated captions is 40. Additionally, the results may be influenced by the challenge of producing discourse cues, such as conjunctions, that indicate the relationship between events (see Appendix G). This can be difficult for captioning techniques. We believe that the 'parsing after captioning' framework may offer certain advantages. However, such findings lead us to conclude that this approach is not a viable option due to the unavailability of human-level captions.

Right Branching achieved remarkable scores for both Span and Nuclearity, while Left Branching obtained significantly lower scores. The performance of Right Branching is superior to that of Video Captioning and Description methods. Since the story of a video is formed by events along a timeline, the left spans tend to depend on the right spans. Thus, it might seem natural that the right-heavy binarized RST tree would obtain good scores for both Span and Nuc.

The results suggest that we should concentrate our efforts on enhancing parsers by leveraging visual features. Low-quality captions that significantly degrade parsing performance come from errors that accumulate in both the encoder and decoder. However, the vector output generated by the encoder could still effectively link knowledge between video and text without being affected by decoding errors. Notably, the parser's encoder and the video captioning model's encoder can share the same architecture; they are both transformer encoders with TSP. Consequently, we can improve the parser by pre-training its encoder with the video captioning task, effectively incorporating text knowledge in the encoder. We performed pre-training of the parser's encoder through Video Captioning and Video Description. The ActivityNet Captions dataset was used for the pre-training, and the results are shown in Table 7. When using Video Description, parsing performances improved for Span, Nuc., and Rel.; however, no improvement was observed for Full. When employing Video Captioning, no improvement was observed. These results imply that the parser's encoder trained with Video Description potentially encodes knowledge regarding the relations between events. This is because it learns to generate captions that result in a coherent storyline for a video, as opposed to individual captions. Our findings suggest that this approach is promising for transferring textual knowledge that generates coherent stories into the parser's vision encoder.

	Span	**Nuc.**	**Rel.**	**Full**
Visual Features	38.3	25.4	19.8	**18.8**
+ Pre-train. w/ VC	38.0	24.1	18.4	17.1
+ Pre-train. w/ VD	**40.5**	**26.4**	**20.2**	18.7

Table 7: Evaluation results using pre-training with video captioning. Scores are the average of five trials with different seeds.

	Tree	**Cap.**	**EA**	**R-1**	**R-2**	**R-L**
Random	–	G	.489	.345	.147	.290
		P		.126	.0360	.161
Lead	–	G	.521	.373	.165	.301
		P		.111	.0360	.157
Tail	–	G	.492	.312	.131	.260
		P		.127	.0354	.159
Even	–	G	.515	.322	.133	.282
		P		.125	.0364	.161
DFS	G	G	.585	.381	.161	.300
		P		.135	.0383	.163
	P	G	.513	.356	.151	.297
		P		.129	.0378	.162
BFS	G	G	.558	.356	.143	.290
		P		.121	.0355	.162
	P	G	.487	.322	.138	.276
		P		.130	.0367	.161

Table 8: Evaluation results of multimodal summarization. G represents gold and P represents predicted tree/captions. The predicted trees were obtained from Visual Features + Pre-train w/ VD, and the predicted captions were obtained from the VD model.

8.4 Experimental Results on Summarization

We compared simple rule-based summarization methods to demonstrate the effectiveness of video RST trees for multimodal summarization. Table 8 shows the results. Random generates a summary by randomly selecting events of the same number in the reference summary (denoted as N events) and extracting the corresponding captions. Lead selects the first N events, Tail selects the last N events, and Even selects N events at even intervals. On the other hand, DFS and BFS select N events using the video RST tree. DFS selects events in a depth-first manner, while BFS selects events in a breadth-first manner. These methods are simple rule-based tree-trimming techniques that utilize the structure of the RST tree to consider relationships between events. Note that the video RST trees are transformed into a dependency format to represent parent-child relationships between events by using rules in (Hirao et al., 2013) (see Appendix H).

From the table, we can see that, when using gold trees, DFS and BFS achieve higher Event Agreement rates than the other methods. Although the performance degrades when using predicted trees, DFS still outperforms Random and Tail, and performs comparably to Even. When employing gold captions, DFS with gold trees achieves the highest ROUGE-1 score, while BFS with gold trees obtains the best ROUGE-L score. Notably, even with predicted trees, DFS achieves the second-best ROUGE-1 score. Despite the significant degradation in ROUGE scores when using predicted captions, DFS and BFS consistently outperform the other methods with both gold and predicted RST trees. Lead and Tail might be effective when the crucial information is concentrated at the beginning or end of the video. However, DFS and BFS have the advantage of extracting frames/captions based on the tree structure that represents the video's main themes and flow, without assuming a bias in the position of important information. This advantage allows DFS and BFS to surpass both Lead and Tail in most cases. Moreover, the fact that they also outperform Even highlights the importance of considering the relationships between events in the video. These findings provide strong evidence that video RST trees play a crucial role in video understanding, just as text RST trees are essential for text understanding.

9 Conclusion

This paper introduced a new task, video discourse parsing, to build an RST tree whose leaf nodes correspond to events in a video. We constructed a dataset, VDTB, consisting of 1,100 videos with high-quality annotations for events, captions, and RST trees, as demonstrated by the comparison with ActivityNet Captions and RST-DT. The evaluation of baseline parsers suggested that improving the parser with visual features and transferring textual knowledge to the vision encoder are promising approaches. Finally, we demonstrated that video RST trees enhance multimodal summarization performance. By employing DFS or BFS to traverse the dependency format of the RST tree and extract keyframes and captions, we achieved better Event Agreement rate and ROUGE scores than other methods, even when using predicted trees. These results highlight the potential of leveraging discourse structures in videos for various downstream tasks that require video understanding.

Limitations

Our dataset comprises only 1,100 annotated videos, which may be insufficient to train neural models effectively. VDTB is smaller than ActivityNet Captions due to the complexity of annotating RST trees, which requires annotators with NLP expertise. The laborious costs associated with such specialized annotation make it challenging to scale to ActivityNet Captions' size.

We also recognize the importance of extending this research to less structured videos. For such cases, alternative discourse parsing approaches like PDTB-style parsing (Prasad et al., 2008), which identifies partial discourse structures, might be more suitable.

In our experiments, we employed gold-standard segmentation to identify event spans. Although it is preferable to automatically identify the event spans for given videos in real-world applications, this poses a critical problem: the leaf nodes of the predicted RST tree may not align with those of the correct RST tree. This misalignment makes Standard-Parseval, a widely used evaluation metric for text RST parsing, unavailable for video RST parsing. To address this issue, we need to develop a new evaluation metric specifically designed for video RST parsing that can handle the misalignment between the predicted and correct RST trees.

References

Arjun R. Akula and Zhu Song-Chun. 2020. Discourse Parsing in Videos: A Multi-modal Appraoch. In *Proceedings of the CVPR Workshop on Language and Vision 2020*.

Humam Alwassel, Silvio Giancola, and Bernard Ghanem. 2020. TSP: Temporally-Sensitive Pretraining of Video Encoders for Localization Tasks. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3166–3176.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory.

In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

William Falcon et al. 2019. PyTorch Lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. SODA: Story Oriented Dense Video Captioning Evaluation Framework. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, pages 517–531.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *Computer Vision – ECCV 2014*, pages 505–520, Cham. Springer International Publishing.

A. Hanjalic, R.L. Lagendijk, and J. Biemond. 1999. Automated high-level movie segmentation for advanced video-retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):580–588.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1515–1520, Seattle, Washington, USA. Association for Computational Linguistics.

Lei Ji, Chenfei Wu, Daisy Zhou, Kun Yan, Edward Cui, Xilin Chen, and Nan Duan. 2022. Learning temporal video procedure segmentation from an automatically collected large dataset. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2733–2742.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2022. A Simple and Strong Baseline for End-to-End Neural RST-style Discourse Parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6754–6766.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pages 706–715.

Hilde Kuehne, Ali Arslan, and Thomas Serre. 2014. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. 2020a. Video storytelling: Textual summaries for events. *IEEE Transactions on Multimedia*, 22(2):554–565.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. 2020b. Video storytelling: Textual summaries for events. *IEEE Transactions on Multimedia*, 22(2):554–565.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81.

Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. SwinBERT: End-to-End Transformers With Sparse Attention for Video Captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17949–17958.

Ilya Loshchilov and Frank Hutter. 2017. Fixing Weight Decay Regularization in Adam. *CoRR*, abs/1711.05101.

Zelun Luo, Wanze Xie, Siddharth Kapoor, Yiyun Liang, Michael Cooper, Juan Carlos Niebles, Ehsan Adeli, and Fei-Fei Li. 2021. Moma: Multi-object multi-actor activity parsing. In *Advances in Neural Information Processing Systems*, volume 34, pages 17939–17955.

Mathieu Morey, Philippe Muller, and Nicholas Asher. 2017. How much progress have we made on RST discourse parsing? a replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1319–1324.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 8024–8035.

Mingtao Pei, Yunde Jia, and Song-Chun Zhu. 2011. Parsing video events with goal inference and intent prediction. In *2011 International Conference on Computer Vision*, pages 487–494.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In

Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08).

Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. 2018. Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4171–4179.

Alexander Richard, Hilde Kuehne, and Juergen Gall. 2017a. Action sets: Weakly supervised action segmentation without ordering constraints. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5987–5996.

Alexander Richard, Hilde Kuehne, and Juergen Gall. 2017b. Weakly supervised action learning with rnn based fine-to-coarse modeling. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1273–1282.

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. 2020. Intra- and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, and Song-Chun Zhu. 2013. Joint video and text parsing for understanding events and answering queries. *IEEE MultiMedia*, 21:42–70.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based Image Description Evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VATEX: A Large-scale, High-Quality Multilingual Dataset for Video-and-Language Research. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4581–4591.

Yasuhiko Watanabe, Yoshihiro Okada, Sadao Kurohashi, and Eiichi Iwanari. 2000. Discourse structure analysis for news video. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, pages 53–60, Centre Universitaire, Luxembourg. International Committee on Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. 2022. Video question answering: Datasets, algorithms and challenges. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6439–6455, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards Automatic Learning of Procedures from Web Instructional Videos. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence*, pages 7590–7598.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.

Wanrong Zhu, Bo Pang, Ashish V. Thapliyal, William Yang Wang, and Radu Soricut. 2022. End-to-end dense video captioning as sequence generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5651–5665, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

A Video Summarization Using RST Trees

Fig. 3 shows example summaries. Summary (a) does not retain the story of the original video, while summary (b) maintains it. Summary (a) illustrates two competitors who performed in the high jump and a lap of honor; one of them was the second competitor. However, it is misleading as it implies that the second competitor won the competition, which is not true. On the other hand, summary (b) shows attempts by three competitors and a lap of honor by the second and third competitors. It can be inferred that the third competitor won the competition, as evidenced by the audience's applause after his attempt.

B Dataset Construction with Two Annotators

We show the procedure for constructing the dataset with two annotators for examining the inter-annotator agreement in Fig. 4.

C Rhetorical Relations

In the following, we show our nine rhetorical relations assigned between two intermediate sibling nodes (see Fig. 5):

BACKGROUND (mono-nuclear relation) is assigned between two intermediate sibling nodes when an event span[12] represents the background of the other event span. An example from ID=174:

Satellite [Students with bouquets and soldiers gather at the airport arrival area.]$_{S_1}$

Nucleus [The students and the soldiers wait in a line with bouquets and wreaths.]$_{S_2}$ [Vice Premier Ro Tu Chol and Vice Minister of Sport Kim Jong Su also wait in a line.]$_{S3}$

CAUSE (mono-nuclear relation) is assigned when an event span causes the other unexpected event span, which frequently appears in videos about activities. An example from ID=1064:

Satellite [In a room, a girl walks toward a dog with a leash to take the dog out of the house.]$_{S_1}$

Nucleus [However, the dog walks away from the girl.]$_{S_2}$

COMPARISON (multi-nuclear relation) is assigned when event spans are compared with each other at equal significance. For example, the relation is used to distinguish winners from losers in sports or games. An example from ID=489:

Nucleus [Canadian curlers show calm faces.]$_{S_8}$

Nucleus [On the other hand, American curlers show frowning faces.]$_{S_9}$

PREPARATION (mono-nuclear relation) is assigned when one event span is in a procedural relation with another event span, such as instructional guidance for a given procedure. An example from ID=025:

Satellite [Next, he locates the car's coolant reservoir.]$_{S_2}$ [And then, he opens the cap of the coolant reservoir.]$_{S_3}$

Nucleus [He inserts a funnel into the reservoir.]$_{S_4}$

RESULT (mono-nuclear relation) is assigned when an event span leads to the other event span without strong causality, such as the results of instructions and temporal changes in events. An example from ID=1061:

Satellite [The actor talks about his previous experience to the homeless man.]$_{S_{11}}$ [Then, the actor gives money to the homeless man.]$_{S_{12}}$

Nucleus [The actor says goodbye to the homeless man.]$_{S_{13}}$

SUPPLEMENT (mono-nuclear relation) is assigned when an event span supplements the other event span. This relation usually appears in human-edited videos to emphasize significant events. An example from ID=001:

Nucleus [Surrounded by audience at a square, the spokesperson of the protesters makes a speech.]$_{S_4}$

Satellite [Some of the audience applaud him.]$_{S_5}$

LIST (multi-nuclear relation) is assigned when event spans are listed at the same significance, such as simultaneous events. An example from ID=670:

Nucleus [A Buddhist monk walks in front of a temple at sunrise]$_{S_1}$

Nucleus [Another Buddhist monk cleans a street.]$_{S_2}$

SUMMARY (mono-nuclear relation) is assigned when an event span summarizes the other event span. A typical example is a digest scene in videos. An example from ID=728:

Satellite [A digest of the video plays.]$_{S_1}$

Nucleus [The bartender talks about the Moscow Mule.]$_{S_2}$ ... [The drink is garnished with a slice of lime and the Moscow Mule is finished.]$_{S_8}$

[12] An event span consists of single or multiple events, and it is dominated by an intermediate node in an RST tree.

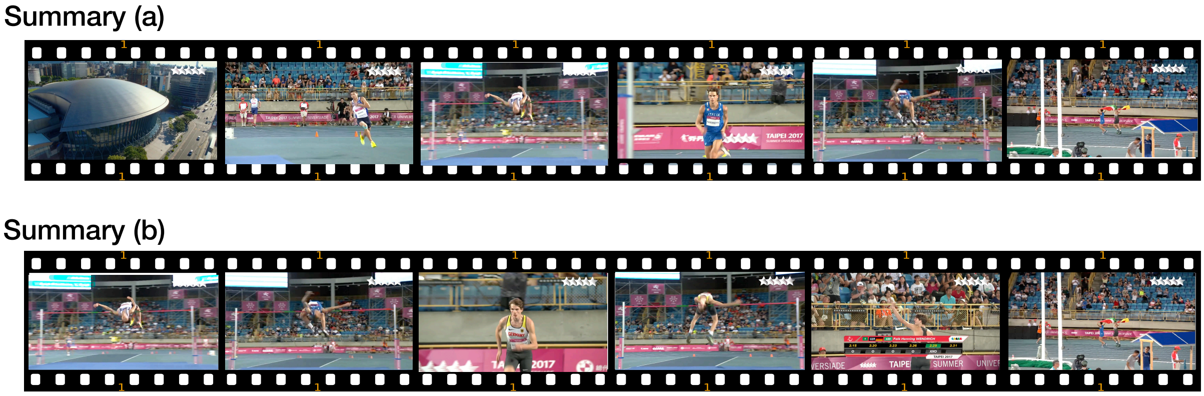


Figure 3: An incoherent summary (a) and a coherent summary (b)

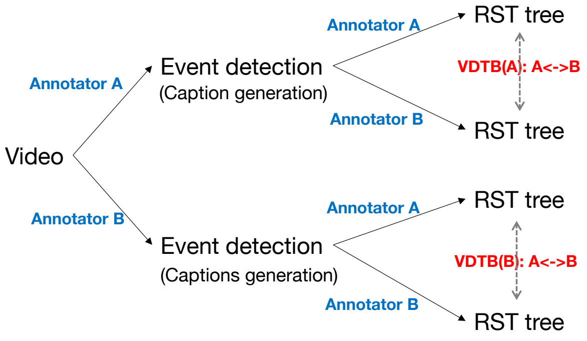


Figure 4: Procedure for dataset construction

Rank	VDTB	RST-DT
1	Preparatin (40.8%)	Elaboration (42.9%)
2	Supplement (26.7%)	Joint (13.4%)
3	List (9.81%)	Explanation (8.29%)
4	Cause (8.45%)	Contrast (6.92%)
5	Background (8.37%)	Evaluation (5.33%)
6	Result (2.53%)	Background (4.54%)
7	Summary (1.83%)	Cause (3.31%)
8	Comparison (0.89%)	Topic-Change (2.44%)
9	Restatement (0.61%)	Temporal (2.18%)
10	–	Attribution (1.92%)
11	–	Textual-ogranization (1.92%)
12	–	Comparison (1.51%)
13	–	Topic-Comment (1.43%)
14	–	Summary (1.40%)
15	–	Same-unit (0.92%)
16	–	Enablement (0.80%)
17	–	Condition (0.67%)
18	–	Manner-Means (0.44%)

Table 9: Distribution of rhetorical relations in VDTB and RST-DT

RESTATEMENT (mono-nuclear relation) is assigned when an event span represents the repetition of the another event span. A typical example is a replay scene in videos. An example from ID=427:

Satellite [The girls show some painted pictures.]$_{S_3}$

Nucleus [The instructor lays a drawing on a canvas.]$_{S_4}$
. . . [They show their finished paintings.]$_{S_8}$

D Distribution of Rhetorical Relations

Table 9 shows the distribution of rhetorical relations in VDTB and RST-DT.

E Implementation Details

We implemented all models based on PyTorch (Paszke et al., 2019) with PyTorch Lightning (Falcon et al., 2019) and used language models from HuggingFace's Transformers (Wolf et al., 2020). The dimension of hidden layers in FFNs was set to 512, and the dropout rate was set to 0.2. The video transformer encoder had 512 embedding dimensions, 8 heads for multi-head attention, 1024 dimensions for the feed-forward layer, and 0.2 for the dropout rate. The batch size was set to 5 actions for text/video RST parsing and 256 captions for Video Captioning and Description. We optimized all models with the AdamW (Loshchilov and Hutter, 2017) optimizer. The learning rate was chosen from {1e-2,1e-3,1e-4,1e-5} using the validation set. Learning rates of 1e-5 and 1e-4 were used for text and video RST parsing, respectively. 1e-3 was used for Video Captioning and Description. We scheduled the learning rate by linear warm-up, which increases the learning rate linearly during the first epoch and then decreases it linearly to 0 until the final epoch. We trained the model for up to 20 epochs and chose the best model based on the evaluation metrics[13] using the validation set. The other hyperparameters used in our experiments are shown in Table 10.

[13] We used the micro-averaged F1 score of Full for the video RST parsing tasks and the BLEU score for the captioning tasks.

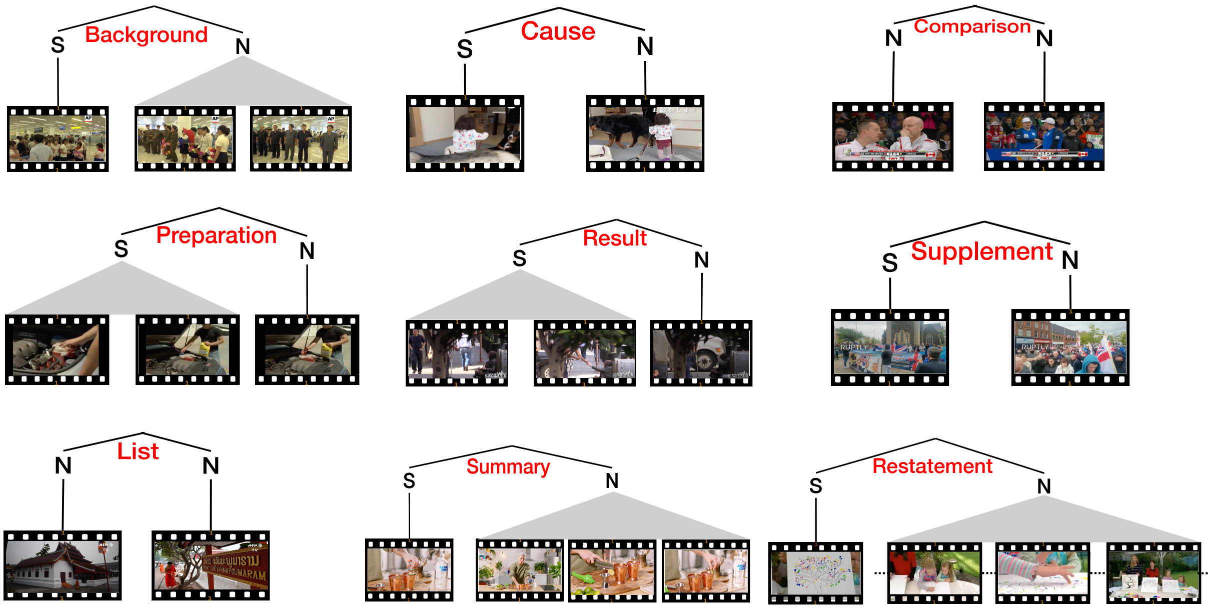


Figure 5: Examples of rhetorical relations in videos

GPU	GeForce RTX 3090
Number of training epochs	20
Batch size (Number of actions)	5
Text encoder	DeBERTa-v3-base
Video encoder	Transformer with TSP (8 heads, 2 layers, 1024 dim's FFN)
FFN's hidden size	512
Dropout	0.2
Learning rate scheduler	Linear warm-up
Optimizer	AdamW
Learning rate	1e-2,1e-3,1e-4,1e-5
Weight decay	0.01
Gradient clipping	1.0
Validation criteria	Standard-Parseval: Full

Table 10: Parameter search space in our text/video RST parsing experiments

	BLEU-4	ROUGE-L	METEOR	CIDEr
SwinBERT	**0.422**	13.1	**7.10**	**19.0**
Video Captioning	0.124	11.9	5.23	11.8
Video Description	0.194	**14.5**	5.23	14.2

Table 11: Evaluation results of automatic captioning on the test set of VDTB. Scores represent the average of five trials with different seeds and are presented as percentages.

F Automatic Video Captioning

To investigate the impact of the caption quality on the 'parsing after captioning' approach for video RST parsing, we first evaluated the performance of the following three captioning methods by comparing their captions with the gold captions:

SwinBERT (Lin et al., 2022) is one of the SOTA video captioning models, trained with VATEX (Wang et al., 2019).

Video Captioning is a transformer-based captioning model trained with ActivityNet Captions (Krishna et al., 2017) originated from (Zhou et al., 2018b), whose objective function is designed to generate a caption for an event.

Video Description is a transformer-based video story generation model trained with ActivityNet Captions, a simplified variant of (Zhu et al., 2022), whose objective function is designed to generate a story for an entire video rather than a single event. Fig. 6 shows the architectures of the Video Captioning and Video Description models used in our experiments. The upper part of Fig. 6 indicates Video Captioning trained to generate a caption for an event. The lower part indicates Video Description trained to generate a sequence of captions for an entire video.

Table 11 shows BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and CIDEr (Vedantam et al., 2015) scores[14] for each method on the test set of VDTB. From the table, SwinBERT performed the best except for ROUGE-L, with significant gains. Video Description surpassed Video Captioning, though marginally.

[14] Note that METEOR scores are identical to SODA scores when using ground truth events.

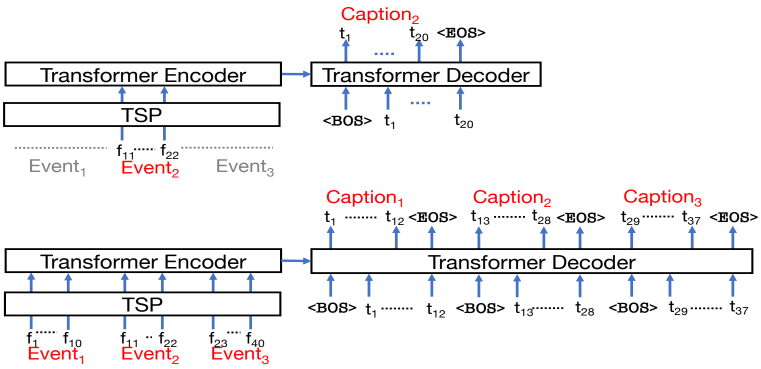


Figure 6: Two video captioning models. The upper and lower parts of the figure show Video Captioning and Description, respectively.

G Example RST Trees

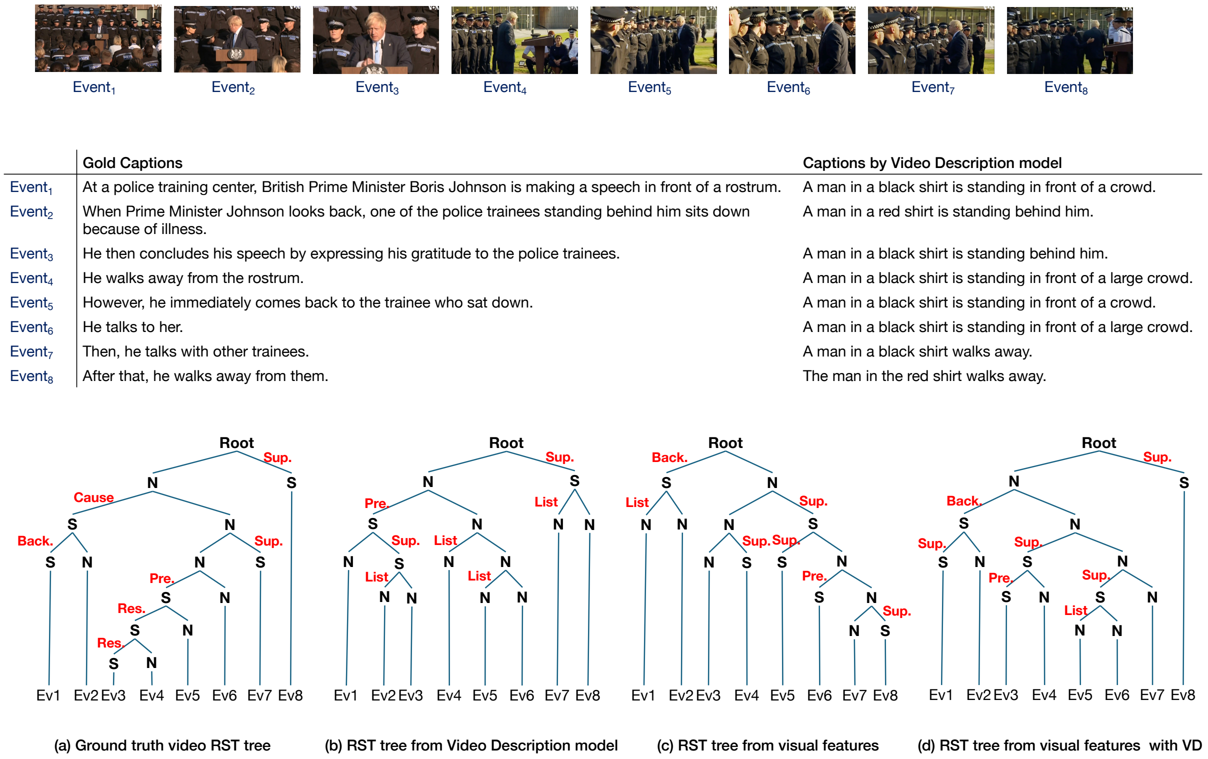
Fig. 7 shows a source video, corresponding captions, and corresponding RST trees. The captions generated by the Video Description model are pretty similar to each other, resulting in the RST tree (b), having multiple nucleus structures representing 'List.' Unfortunately, this tree was far from the ground truth (a) and had a Span score of 0. On the other hand, the RST tree (c), obtained from visual features, had fewer multiple nucleus structures, but it was still distinct from the ground truth, with a Span score of 28.5. However, applying transfer learning to the visual feature parser significantly improved the output, resulting in the RST tree (d), with several subtrees matching those of the ground truth RST tree. This enhancement led to a Span score of 71.4.

H Conversion from RST Trees to Dependency Trees

Fig. 8 shows an example RST tree obtained from a video and its corresponding dependency format. We can convert RST trees into dependency trees by using the following procedure: (1) For any given event, find the nearest satellite (S) among its ancestors. (2) From the sibling nucleus (N) of the nearest satellite, follow only the rightmost nuclei downward in the tree until reaching an event. (3) Assign the event found in step (2) as the parent of the event from step (1). Note that if no satellite is found among the ancestors of an event, the event reached by following only nuclei from the root node is assigned as its parent. Furthermore, if the assigned parent of an event is the event itself, then that event becomes the root node of the dependency tree.

In the figure, we can see that the parent of Event$_3$ is Event$_5$, which is determined by applying this procedure. Additionally, if we consider Event$_4$, since it has no satellite ancestor, its parent would be Event$_5$, the event reached by following only nuclei starting from the root node. Finally, Event$_5$ is the root node of the dependency tree because its assigned parent is itself.

DFS and BFS traverse the dependency trees in a depth-first and breadth-first manner, respectively, to choose events. When we set N to three, i.e., we select three events from a video, DFS extracts Event$_2$, Event$_3$, and Event$_5$, while BFS extracts Event$_3$, Event$_4$, and Events$_5$.



	Gold Captions	Captions by Video Description model
Event$_1$	At a police training center, British Prime Minister Boris Johnson is making a speech in front of a rostrum.	A man in a black shirt is standing in front of a crowd.
Event$_2$	When Prime Minister Johnson looks back, one of the police trainees standing behind him sits down because of illness.	A man in a red shirt is standing behind him.
Event$_3$	He then concludes his speech by expressing his gratitude to the police trainees.	A man in a black shirt is standing behind him.
Event$_4$	He walks away from the rostrum.	A man in a black shirt is standing in front of a large crowd.
Event$_5$	However, he immediately comes back to the trainee who sat down.	A man in a black shirt is standing in front of a crowd.
Event$_6$	He talks to her.	A man in a black shirt is standing in front of a large crowd.
Event$_7$	Then, he talks with other trainees.	A man in a black shirt walks away.
Event$_8$	After that, he walks away from them.	The man in the red shirt walks away.

(a) Ground truth video RST tree (b) RST tree from Video Description model (c) RST tree from visual features (d) RST tree from visual features with VD

Figure 7: A source video, corresponding captions, corresponding RST trees obtained from different parsing models

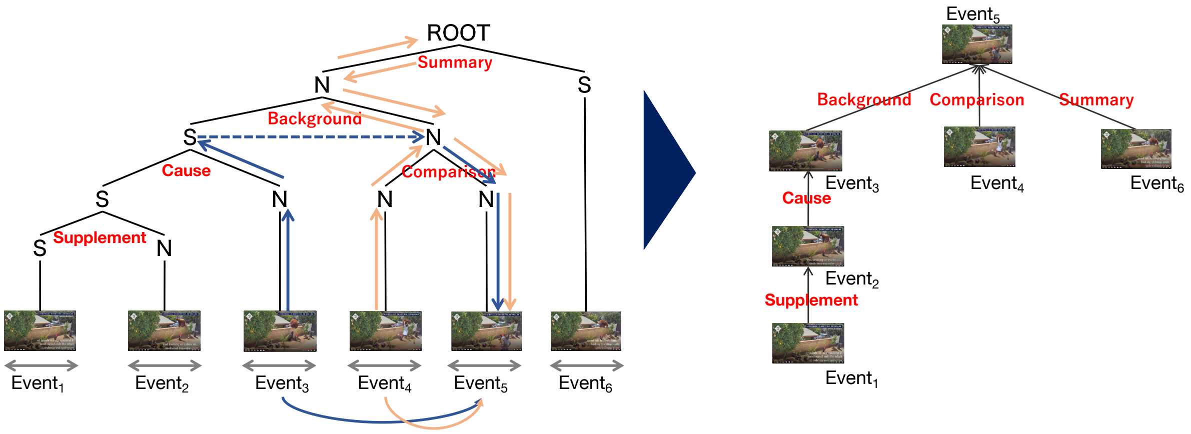


Figure 8: An RST tree and corresponding dependency tree