End-to-end Dense Video Captioning as Sequence Generation

Wanrong Zhu[1], Bo Pang[2], Ashish V. Thapliyal[2], William Yang Wang[1], Radu Soricut[2]
[1]UC Santa Barbara, [2]Google Research
{wanrongzhu,william}@cs.ucsb.edu {bopang,asht,rsoricut}@google.com

Abstract

Dense video captioning aims to identify the events of interest in an input video, and generate descriptive captions for each event. Previous approaches usually follow a two-stage generative process, which first proposes a segment for each event, then renders a caption for each identified segment. Recent advances in large-scale sequence generation pretraining have seen great success in unifying task formulation for a great variety of tasks, but so far, more complex tasks such as dense video captioning are not able to fully utilize this powerful paradigm. In this work, we show how to model the two subtasks of dense video captioning jointly as *one* sequence generation task, and simultaneously predict the events and the corresponding descriptions. Experiments on YouCook2 and ViTT show encouraging results and indicate the feasibility of training complex tasks such as end-to-end dense video captioning integrated into large-scale pretrained models.

1 Introduction

Online videos have become an important source of knowledge and skills (O'Neil-Hart, 2017). In order to help users locate information of interest, search engines and video platforms often show anchors at "key moments", usually accompanied by descriptions of the segment's content (Baheti, 2019). This is a direct application of the dense video captioning task (Krishna et al., 2017), thus methods for improving performance on this task are relevant to any video platform.

Intuitively, dense video captioning can be decomposed into two subtasks: event localization and segment-level video captioning. Following this approach, prior work (Krishna et al., 2017; Zhou et al., 2018a; Li et al., 2018; Wang et al., 2018; Zhou et al., 2018c; Mun et al., 2019; Iashin and Rahtu, 2020) has used a two-stage, "localize-then-describe" pipeline. Such methods usually involve two separate modules with different underlying model architectures for event localization and event captioning, with captions for dense events rendered based on the predicted event spans.

Recently, with the advance of large-scale datasets and model architectures, there has been an explosion of pretrained multimodal (for text, image, video) Transformer models (Tan and Bansal, 2019; Sun et al., 2019; Li et al., 2019; Luo et al., 2020; Li et al., 2020a,b; Gan et al., 2020; Kim et al., 2021). Such models have proved to be highly effective when fine-tuned for a wide range of downstream tasks, such as visual question answering (Agrawal et al., 2015), image captioning (Chen et al., 2015), visual common sense reasoning (Zellers et al., 2019), visual entailment (Xie et al., 2019), etc. These end-tasks can be expressed as sequence generation tasks in a straightforward manner. In contrast, this is non-trivial for dense video captioning, as the segmentation subtask does not lend itself naturally to such a formulation. Does this mean more complex tasks cannot benefit from the pretraining paradigm in an end-to-end fashion? In this work, we study dense video captioning as an example of a complex task that can be cast as sequence generation and, as a result, can benefit from large-scale pretraining.

More specifically, we propose to solve the dense video captioning task as a single sequence-to-sequence modeling task using a multimodal Transformer. To this end, we design several task formulations to encode both segmentation and caption prediction in one target string. Thus, our task formulations allow the model to simultaneously predict event locations and corresponding captions in one pass, using one decoder. This opens the door to leveraging large-scale pretrained models, as well as the option of participating in large-scale multi-task training more easily by reusing existing infrastructure.

We evaluate our model on two dense video

Proceedings of the 29th International Conference on Computational Linguistics, pages 5651–5665
October 12–17, 2022.

captioning benchmarks, YouCook2 (Zhou et al., 2018a) and ViTT (Huang et al., 2020a). Our sequence generation formulations provide a feasible path forward – we obtain encouraging results compared to prior work that used a two-stage scheme with specialized architectures for each step. On the pretraining front: (a) we are able to benefit from models pretrained on very different data and tasks, such as T5 (Raffel et al., 2020), (b) pretraining on more domain-specific data (WikiHow) and pretraining tasks (predicting headings for how-to steps) lead to a similar amount of gain, but (c) having the domain-specific pretraining start from a T5 checkpoint (T5 + WikiHow) provides a significantly larger gain. The noteworthy result is that, even in the presence of large-scale domain- and task-specific pretraining (WikiHow), one can still observe measurable benefits from a task-agnostic general-purpose pretrained model (T5).

While the primary motivation for modeling the two tasks jointly is to be able to utilize the pretraining paradigm, the segmentation subtask (finding event boundaries) and the captioning subtask (describing what happens in an event) are related tasks, and intuitively stand to benefit from being modeled jointly. Our experimental results are aligned with this intuition: a model that does both segmentation and captioning simultaneously, outperforms (in terms of segmentation accuracy) a variant that focuses only on the segmentation task.

Overall, our results point to a viable alternative direction for modeling complex tasks such as end-to-end dense video captioning, in which we can leverage the large-scale pretraining paradigm to achieve modeling improvements.

2 Related Work

2.1 Multimodal Transformer

Recently, vision-and-language pre-training has attracted a lot of attention for jointly learning from visual and textual inputs in order to better solve multimodal tasks. Following the success of BERT (Devlin et al., 2019), multimodal pre-training usually adopts the Transformer (Vaswani et al., 2017) encoder structure to encode both the visual features and textual features. The late-fusion approaches first process visual and textual information separately and subsequently fuse them using another Transformer layer (Tan and Bansal, 2019; Lu et al., 2019). The early-fusion approaches jointly encode visual and texual representations (Chen et al., 2020; Sun et al., 2019; Li et al., 2019; Luo et al., 2020; Li et al., 2020a; Qi et al., 2020; Huang et al., 2020b; Li et al., 2020b; Lin et al., 2020; Gan et al., 2020; Kim et al., 2021). During pre-training, tasks such as masked language modeling, masked region modeling, and image-text matching are used to learn a cross-modal encoding which benefits downstream multimodal tasks.

2.2 Dense Video Captioning

Krishna et al. (2017) introduced the dense video captioning (DVC) task and proposed a solution based on two separate modules: one for proposing events, and another for captioning them. Recent work (Zhou et al., 2018a; Li et al., 2018; Wang et al., 2018; Zhou et al., 2018c; Mun et al., 2019; Iashin and Rahtu, 2020) follows the two-stage “detect-then-describe” framework, in which the event proposal module first predicts a set of event segments, then the captioning module constructs captions for each candidate event segment. Another line of work (Deng et al., 2021; Wang et al., 2021) removes the explicit event proposing process. Deng et al. (2021) tackles the DVC task from a top-down perspective, in which they first generate a video-level story, then ground each sentence in the story into a video segment. Wang et al. (2021) considers the DVC task as a set prediction problem, and applies two parallel prediction heads for event localization and captioning. To the best of our knowledge, our work is the first to simultaneously conduct event localization and captioning in a single run[1] within the same prediction head for the dense video captioning task.

3 Task Definition

The DVC task consists of annotating each input video into multiple segments, where each segment corresponds to an event of interest accompanied by a short description (caption). Figure 1 shows an example from the YouCook2 dataset.

Modified dense video captioning In YouCook2, each segment is marked by a start and an end time, often with gaps between segments. The burden of identifying not just the right start-time but also the right end-time increases the difficulty of the

[1] Note that on a different task (object detection), contemporaneous work (Chen et al., 2022) has combined spatial localization and object description via a sequence generation formulation by predicting bounding box coordinates and object labels in sequence.

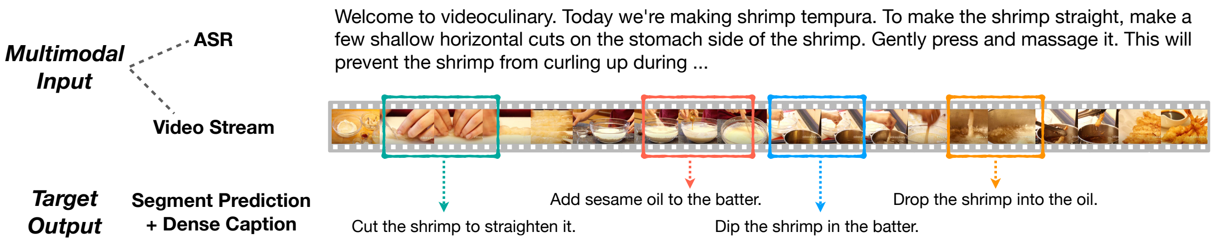

Multimodal Input
ASR
Welcome to videoculinary. Today we're making shrimp tempura. To make the shrimp straight, make a few shallow horizontal cuts on the stomach side of the shrimp. Gently press and massage it. This will prevent the shrimp from curling up during ...
Video Stream
Target Output
Segment Prediction + Dense Caption
Cut the shrimp to straighten it.
Add sesame oil to the batter.
Dip the shrimp in the batter.
Drop the shrimp into the oil.

Figure 1: An example of the input video and output segmentations and captions from the YouCook2 dataset.

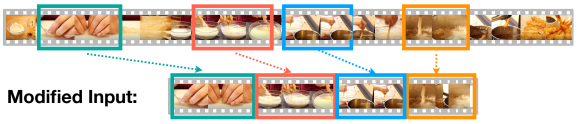


Figure 2: Modified dense video captioning: a simplified setting where the segments are concatenated to form the modified input with gaps removed. Table 1, 5, 6 show our preliminary experiments using this set up; all other results reported in the paper are carried out in the original setting as depicted in Figure 1.

segmentation task. Thus, we start our exploration with a simpler task where we introduce a variant of the YouCook2 dataset as shown in Fig. 2: all the annotated segments in a given video are concatenated to form a *modified* input, leaving out the gaps between segments. We refer to this setting as the *modified dense video captioning*: given an input from Fig. 2, the model only needs to predict n start times to fully define n segments. In this setting, the segmentation subtask becomes a *partition* task for identifying the set of start times of segments.

4 Method

As noted earlier, prior work often decomposes dense video captioning into two subtasks, (a) a segmentation subtask, and (b) a segment-level captioning subtask. These two subtasks are often addressed with different model architectures. In contrast, our approach solves both subtasks simultaneously with one single model.

We first describe how we jointly model segmentation and captioning subtasks as one single sequence generation task. To this end, we need to formulate target strings in ways that encode both segmentation and captioning predictions.

The typical input to a DVC task includes both visual information and speech in textual form – Automatic Speech Recognition (ASR) tokens. We start by introducing our target string formulations assuming only textual input, with segmentation information expressed in terms of the positions of the corresponding ASR tokens[2]. We then describe multi-modal models where the visual information is added to the input while retaining the aforementioned scheme to represent segmentation information.

4.1 Target string formulations

We describe two approaches to formulate the target strings. We refer to a model that encodes only segmentation information in the target strings as a **Seg-only model**, and one that encodes both segmentation and captioning as a **Seg+Cap model**.

Tagging-based target formulation We encode the segmentation subtask in a manner similar to the encoding of the chunking task as tagging tokens in the IOB format (Ramshaw and Marcus, 1995). Fig. 3 illustrates how we model the segmentation task with two tags (in the modified setting): the ASR token at the start of a segment receives a special token ⟨sep⟩ as the start-of-segment tag, and the rest of the tokens in the segment receive a continuation tag (we reuse the ⟨pad⟩ token). This can be extended to cover the original setting (with gaps between segments) with an additional end-of-segment tag. In this formulation, the ground-truth target output string has the exact same length as the input ASR string. To model the captioning annotation, the ⟨sep⟩ token is followed by the corresponding ground-truth caption, which is then padded till the next ⟨sep⟩ token.

While treating the segmentation task as a tagging task seems natural, the tagging-based formulation enforces equal lengths between predicted output and the input ASR tokens, which leads to potential inefficiencies: the input ASR string is usually much longer than all the descriptive captions combined, which results in many padding tokens in the target output, and leads to an unnecessary slow-down

[2] Our motivation for treating DVC as a sequence generation task is to take advantage of existing pretrained sequence generation models, currently dominated by text models; thus, we take a text-centric view in this work.

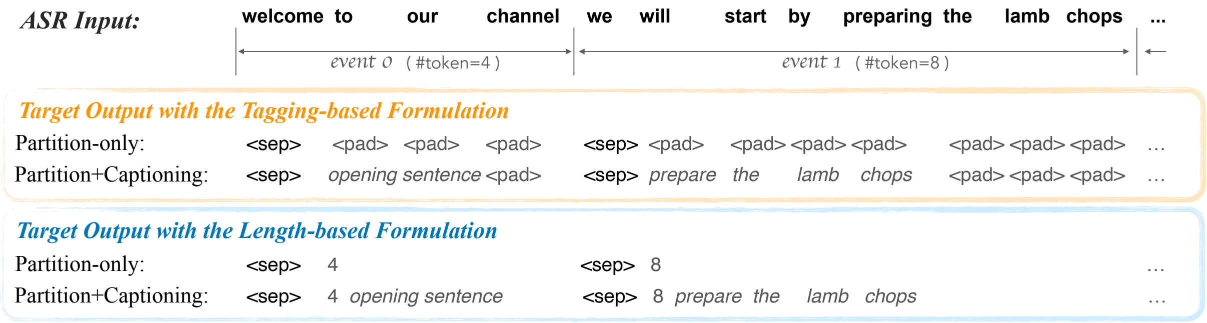

ASR Input:
welcome to our channel we will start by preparing the lamb chops ...
event 0 (#token=4)
event 1 (#token=8)
Target Output with the Tagging-based Formulation
Partition-only: <sep> <pad> <pad> <pad> <sep> <pad> <pad> <pad> <pad> <pad> <pad> <pad> …
Partition+Captioning: <sep> opening sentence <pad> <sep> prepare the lamb chops <pad> <pad> <pad> …
Target Output with the Length-based Formulation
Partition-only: <sep> 4 <sep> 8 …
Partition+Captioning: <sep> 4 opening sentence <sep> 8 prepare the lamb chops …

Figure 3: The tagging-based and length-based target formulations for modified dense video captioning.

in training and prediction time. Additionally, the longer-form target strings are markedly different from the usual generative pattern of the pretrained text decoder, which can reduce the effectiveness of the pretrained checkpoints. Furthermore, this formulation also assumes that captions are shorter than the ASR string for each segment; while this is mostly true, for segments where little is being explained (short ASR string), this formulation leaves insufficient capacity in the target string between the two consecutive ⟨sep⟩ tags to encode the corresponding caption, resulting in caption truncation.

Length-based target formulation To cope with the limitations of the tagging-based formulation, we predict the *length* of each segment explicitly.

Let l_i be the number of ASR tokens in the i-th segment. In the modified setting, the segmentation information for an input string with n segments is fully specified by the sequence $\{l_1, l_2, ..., l_n\}$. Fig. 3 provides an example of this length-based formulation. The ground-truth target string in a Seg-only model is simply a sequence of numbers corresponding to segment lengths (measured by the number of tokens); in a Seg+Cap model, each number is followed by the caption for that segment.

In the original setting with gaps between segments, let g_i be the offset from the last ASR token in the previous segment to the start of segment i. The target string will now aim to predict both (g_i, l_i) instead of just l_i for each segment. The sequence of all (g_i, l_i) will fully specify all segment boundaries and can be used to compute the index of the start and end ASR tokens for each segment.

This formulation has the advantage of a more efficient representation of the segmentation information, and thus a much shorter target length. The segmentation information is now explicitly expressed as numbers in the target strings, so the model needs to both figure out segmentation boundaries *and* be able to count appropriately. We explicitly want to empirically measure the ability of our models to do the latter.

4.2 Input formulation for multimodal signals

Simple Concatenation (SimpleConcat) Visual information for a given video is represented as a fixed-length sequence of pre-computed frame-level features. These features are projected to the token embedding space via a fully connected layer. We simply concatenate the sequence of ASR token embeddings and the sequence of projected visual features to form the multimodal input to the encoder. There's one potential caveat: while the visual features are extracted at a fixed frame rate, the ASR tokens are often *not* spoken at a fixed speed; thus positions in this multimodal input sequence do not provide straightforward information on which visual frames are temporally aligned with a certain ASR span. Since segmentation prediction is expressed relative to the ASR-token position index, it is not clear whether the model is able to take full advantage of visual information, absent how these two modalities align temporally.

Prior work on multimodal pretraining has found visual-textual information alignment to be a reasonably solvable task. Huang et al. (2020a) reported 87% accuracy for aligning video segments and ASR spans in HowTo100M (Miech et al., 2019), so it is possible that the decoder can learn to attend to appropriate visual information while "counting" the ASR tokens.

Temporal embedding (Emb$_{\text{TIME}}$) We can also express the temporal alignment more explicitly in the input by adding temporal embeddings to both ASR tokens and visual frames. In this formulation, we learn a temporal embedder shared between the text modality and the visual modality, which maps timestamps to temporal embeddings. Embeddings computed from token timestamps are then added to ASR token embeddings, and embeddings computed from frame timestamps are added to projected visual frame features. This way, ASR tokens and frames that are temporally close to each other receive similar temporal embeddings, making their representations closer to each other.

For more explorations on explicitly expressing temporal alignment in the input, see Appendix A.1 for an additional method to insert explicit timestamp markers into the input text sequence.

5 Experiments

5.1 Datasets

Dense Video Captioning Datasets We use two publicly available datasets to verify the effectiveness of our model formulations: YouCook2 (Zhou et al., 2018a) and ViTT (Huang et al., 2020a).[3] The YouCook2 dataset is restricted to videos retrieved from YouTube from the cooking domain, targeting 89 recipes; each event segment is manually marked with a start and end time, along with a human-generated caption for each tightly-bounded segment. The ViTT dataset contains instructional videos from YouTube-8M (Abu-El-Haija et al., 2016) and covers a broader range of topics. Its segment annotation focuses on event start time and rater-provided captions for the corresponding segment (spanning two consecutive start-time annotations). Both datasets are annotated with captions written in English.

Note that while the YouCook2 data release contains training, dev, and test sets, its test set does not come with human annotations. Thus, we split the original validation set into validation and test splits for our experiments. For ViTT, we use the original train/val/test splits provided with the data. The number of videos available for use at the time of our work[4] for Youcook2 is 925 for train, 206 for validation and 105 for test. For ViTT there are 4736 train, 932 validation and 932 test videos.

Domain-specific pretraining with WikiHow In addition to general-purpose pre-trained models like T5, we also experiment with domain-specific pretraining. To this end, we use the WikiHow dataset (Koupaee and Wang, 2018). WikiHow consists of instructional (how-to) articles, which makes it *in-domain data* for the two dense video captioning datasets considered here, while being much larger in size[5]. In addition, WikiHow articles contain detailed step-by-step instructions. Each step comes with a summary, which usually serves as the section title. Both the step boundaries and summaries are easily extracted according to the page meta-data. This provides the ground-truth annotation for a "dense document caption" task: given the full article as a sequence of text tokens, predict the step boundaries and summaries. This enables us to also include a *domain-specific pretraining task* that closely resembles our task. For each formulation described in Sec. 4, we experiment with a checkpoint pre-trained on the WikiHow data using the corresponding target string formulation.

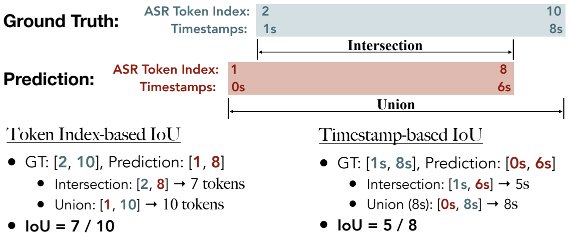

Ground Truth:
ASR Token Index:
Timestamps:
2
1s
10
8s
Intersection
Prediction:
ASR Token Index:
Timestamps:
1
0s
8
6s
Union
Token Index-based IoU
GT: [2, 10], Prediction: [1, 8]
Intersection: [2, 8] → 7 tokens
Union: [1, 10] → 10 tokens
IoU = 7 / 10
Timestamp-based IoU
GT: [1s, 8s], Prediction: [0s, 6s]
Intersection: [1s, 6s] → 5s
Union (8s): [0s, 8s] → 8s
IoU = 5 / 8

Figure 4: Comparisons between the token index-based and timestamp-based IoU used in our study.

5.2 Evaluation Metrics

Segmentation Performance Following previous works (Zhou et al., 2018b; Shi et al., 2019), we use the mean Intersection-over-Union (mIoU) metric to evaluate the segmentation performance. Recall that the ground-truth segments are marked by start (and end) times, whereas the predicted segments are expressed according to the position of the corresponding ASR token. For the modified dense video captioning task, we compute the token index-based IoU: each ground-truth segment is defined by the start and end ASR token index, and will be compared against the predicted index. For the original task, we compute the timestamp-based IoU: predicted indices are mapped into the corresponding ASR token timestamps and compared against the segment's ground-truth start and end timestamps. Fig. 4 provides an example of the two types of IoU used in this study.

An IoU score can be computed for each (ground-truth, predicted) segment pair. The mIoU measure provides a summary score for segmentation performance over the entire video. For each ground-truth segment, we take its maximal IoU to predicted

[3] YouCook2 released under an MIT license; ViTT released under an "AS IS" license.

[4] As of 2021; note that YouTube videos are subject to user deletion.

[5] WikiHow has 157,116 articles in its training set, and 5,593 articles in its validation set.

*	Target Formulation	Checkpoint	Seg-only model				Seg+Cap model					
			mIoU	Precision	Recall	F1	mIoU	F1	B@4	METEOR	CIDEr	ROUGE-L
0	Random Partition		37.74	26.13	24.88	23.52	-	-	-	-	-	-
1	Tagging-based	-	33.59	23.04	29.37	24.46	19.72	17.09	0.07	0.91	0.03	2.07
2		T5	12.06	1.78	7.46	2.81	6.73	0.24	0.00	0.01	0.00	0.03
3	Length-based	-	36.30	26.23	28.79	25.81	33.62	24.69	0.24	1.62	0.04	4.03
4		T5	42.71	31.85	33.04	31.21	42.82	32.16	1.83	4.17	0.21	8.74

Table 1: Preliminary experiments comparing the tagging-based and the length-based formulation on YouCook2 modified dense video captioning. We report the evaluation results on the validation set (one run per setting) with models initialized from random weights or from T5 checkpoints.

segments as the IoU score for this ground-truth segment, and mIoU is the average of this value across all ground-truth segments. The individual mIoU for each video is then averaged across the test data and reported as the overall mIoU.

For diagnostic purposes, we also compute: 1) the percentage of predicted segments which have an IoU score with at least one ground-truth segment above a certain threshold t (precision@t); 2) the percentage of ground-truth segments which have an IoU score with at least one predicted segment, above a certain threshold t (recall@t), as well as their geometric mean as F1. Following prior work, we compute these scores for a set of IoU thresholds t=$\{0.3, 0.5, 0.7, 0.9\}$, and report the average over these thresholds.

Captioning Performance We compute BLEU-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and ROUGE (Lin, 2004) scores between generated captions and the ground truth when the predicted and ground-truth segment "match" (i.e., with IoU score above a given threshold t); if a ground truth segment does not have a matching prediction, it contributes a zero to the average score for the corresponding threshold. Again, we compute this for a set of IoU thresholds of $\{0.3, 0.5, 0.7, 0.9\}$, and report the average over these thresholds.

5.3 Implementation Details

Models were trained on 4x4 TPUs, and we used about 180k GPU hours for around 1380 training runs, including pretraining the WikiHow checkpoint, pilot studies with toy examples, debugging, and hyperparameter tuning. The models have approximately 70 million parameters. We used the Adafactor (Shazeer and Stern, 2018) optimizer and a learning rate schedule of 1000 warmup steps followed by square-root decay. We did a few initial exploratory runs over base learning rates of $\{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 1, 2, 5\}$ to determine that a base learning rate of 1 worked well and used it for all the experiments reported.

For our visual representations, we computed 3D CNN features pre-trained on the Kinetics (Carreira and Zisserman, 2017; Kay et al., 2017) dataset for frames sampled at 30 fps, resulting in one feature for each 1 second clip.

5.4 Experiments in the modified setting

Experimental setup We conduct comparisons of the two different target formulations, tagging-based and length-based, in the modified setting, using the following experimental setup: (a) max input text length and target length are set to 1024, and max input visual feature length is set to 800; this can truncate longer ASR sequences, but allow us to quickly iterate through different settings with fewer computational resources; (b) only one run for each setting. We report results on the validation set in Table 1.

Target formulations The best performing model (row #4 in Table 1: length-based with T5 checkpoint) outperforms a random partition baseline[6] (row #0 in Table 1), which indicates our target formulation approach to the segmentation task is capturing some segmentation information effectively.

When trained from scratch, the length-based formulation achieves higher performance across the board (#3 vs #1), with a smaller gap for the Seg-only model, and a more marked lead for the Seg+Cap model. We hypothesize that while treating the segmentation task as a tagging task is more or less feasible on its own, combining segmentation tags and captions is not a suitable formulation for

[6] For the random partition baseline, a video is randomly partitioned into n segments, with n sampled uniformly from 1 to 15 (The mean number of segments in the ground-truth is 8).

Dataset	*	Input Formulation	Checkpoint?	Seg-only model	Seg+Cap model				
				mIoU	mIoU	B@4	METEOR	CIDEr	ROUGE-L
Youcook2	0	Random Segmentation		20.61 ± 1.04	-	-	-	-	-
	1	SimpleConcat	-	12.99 ± 1.55	16.45 ± 8.72	0.17 ± 0.11	0.66 ± 0.04	0.02 ± 0.01	1.99 ± 0.20
	2		T5	24.14 ± 1.07	24.21 ± 1.64	0.88 ± 0.04	1.50 ± 0.12	0.09 ± 0.01	3.34 ± 0.27
	3		WikiHow	22.58 ± 1.09	23.33 ± 0.79	0.67 ± 0.15	1.47 ± 0.05	0.08 ± 0.01	3.51 ± 0.13
	4		WikiHow T5	**27.77** ± 0.09	**30.26** ± 1.24	**2.96** ± 0.28	**3.49** ± 0.30	**0.25** ± 0.03	**7.00** ± 0.42
	5	+ Emb$_{\text{TIME}}$	-	18.51 ± 1.95	18.71 ± 0.17	0.12 ± 0.07	0.48 ± 0.08	0.02 ± 0.01	1.41 ± 0.22
	6		T5	23.02 ± 1.05	23.96 ± 0.08	1.32 ± 0.08	1.91 ± 0.07	0.11 ± 0.01	4.20 ± 0.13
	7		WikiHow	21.68 ± 1.93	21.88 ± 0.86	0.69 ± 0.19	1.30 ± 0.07	0.07 ± 0.01	3.06 ± 0.13
	8		WikiHow T5	26.51 ± 0.45	28.70 ± 0.92	2.58 ± 0.19	3.23 ± 0.10	0.22 ± 0.01	6.45 ± 0.17
ViTT	9	Random Segmentation		21.90 ± 0.15	-	-	-	-	-
	10	SimpleConcat	-	33.85 ± 0.70	32.69 ± 0.71	0.11 ± 0.01	3.76 ± 0.35	0.08 ± 0.01	3.86 ± 0.28
	11		T5	37.89 ± 0.10	38.07 ± 0.65	0.57 ± 0.03	5.92 ± 0.37	0.16 ± 0.02	6.59 ± 0.69
	12		WikiHow	38.20 ± 0.27	37.80 ± 0.62	0.40 ± 0.07	5.48 ± 0.18	0.14 ± 0.01	6.02 ± 0.34
	13		WikiHow T5	**41.87** ± 0.26	42.40 ± 0.30	**1.29** ± 0.07	**8.10** ± 0.34	**0.25** ± 0.01	**9.26** ± 0.39
	14	+ Emb$_{\text{TIME}}$	-	33.89 ± 0.21	35.37 ± 3.18	0.04 ± 0.03	3.42 ± 0.61	0.07 ± 0.01	3.28 ± 0.83
	15		T5	37.78 ± 0.15	38.50 ± 0.55	0.75 ± 0.10	6.37 ± 0.39	0.18 ± 0.01	7.19 ± 0.48
	16		WikiHow	37.27 ± 0.08	36.97 ± 0.48	0.38 ± 0.06	5.31 ± 0.06	0.13 ± 0.01	5.82 ± 0.23
	17		WikiHow T5	**41.64** ± 0.12	**43.22** ± 0.72	**1.22** ± 0.08	**8.05** ± 0.20	**0.25** ± 0.01	**9.18** ± 0.45

Table 2: Dense video captioning performance on YouCook2 and ViTT test sets with the length-based formulation. We ran 3 trials for each setting, and report the evaluation results (mean ± std) with models initialized from random weights, T5 checkpoints, WikiHow checkpoints, and T5 checkpoints further pretrained on WikiHow. Note: Seg stands for the segmentation task, and Cap stands for the captioning task.

the combined task – to the point that the Seg+Cap model underperforms Seg-only model in segmentation metrics (mIoU of 19.72 vs. 33.59 in #1).

The length-based formulation overall benefits from the T5 checkpoint (#3 vs #4 in Table 1) across different sub-tasks. Note that for the Seg-only model, the target strings (sequences of numbers) are not typically seen in T5 pretraining, but the T5 checkpoint still boosts its performance. In contrast, the tagging-based formulation is not able to benefit from the T5 checkpoint in our experiments. One possible explanation is that the target strings in tagging-based (with large chunks of padding tokens) are just too different from the T5 pretraining targets.

Given the results obtained in the modified setting, we focus our efforts on using length-based target formulation in the more challenging original setting in the following section.

Ablation studies We conducted ablation studies on input modalities, and find models that take text-only inputs stand to benefit more from the pre-trained checkpoints than the models that only take visual inputs. We also conducted ablation studies on partial parameter initialization, and found that partially loading checkpoints from pre-trained models does not work as well as fully loading checkpoints for both the encoder and the decoder. See Appendix (A.2) for more details.

5.5 Experiments in the original setting

Experimental setup Using the length-based target formulation, we conduct a more extensive comparison of the effect of different pretraining strategies, as well as different input formulations on the original dense video captioning task on both YouCook2 and ViTT. Maximum sequence lengths are set to ensure no truncation happens in either dataset – input text: 4096; visual feature: 800 (YouCook2) / 500 (ViTT); target: 512 (YouCook2) / 256 (ViTT). We ran each experiment with different seeds three times to account for performance variance from random initializations. We report the mean and standard deviation (using 3 runs) for each metric in Table 2. We choose the best checkpoint based on performance on the validation set and report the performance on the test set.

Effects of Pretraining For both datasets, there are significant performance improvements from utilizing pre-trained checkpoints in terms of both segmentation metrics and captioning metrics. Interestingly, training from the WikiHow checkpoint (using in-domain task over in-domain data) provides similar performance improvement to T5 alone (see, for instance, #2 vs #3, or #11 vs #12 in Table 2). However, starting from the generic-language T5 checkpoint and adding in-domain WikiHow pretraining (WikiHow T5, e.g., #4 and #13) boosts all metrics by a large and significant margin.

Model	mIoU	Prec.	Rec.	B@4	M
vsLSTM (Zhang et al., 2016)	32.2	24.1	22.1	-	-
SCNN-prop (Shou et al., 2016)	26.7	23.2	28.2	-	-
ProcNet (Zhou et al., 2018b)	37.0	30.4	37.1	-	-
Bi-LSTM + TempoAttn (Zhou et al., 2018c)	-	-	-	0.08	4.62
End2end Transformer (Zhou et al., 2018c)	-	-	-	0.30	6.58
Context-aware Fusion (Shi et al., 2019)	41.4	-	-	2.61	17.43
End2end Sequence Generation (Ours)	30.3	24.5	24.2	2.96	3.49

Table 3: Dense video captioning performance on YouCook2 in the context of prior work. Following prior work, the segmentation performance is measured by the mIoU, the precision (Prec.) and recall (Rec.) at IoU threshold t=0.5. Captioning performance is measured by the average BLEU-4 (B@4) and METEOR (M) at IoU thresholds $t \in \{0.3, 0.5, 0.7, 0.9\}$.

Effects of Joint Modeling If we compare the mIoU score achieved by the Seg+Cap model to the mIoU score achieved by the Seg-only model in Table 2, across different settings, we observe a general trend where the Seg+Cap model outperforms the Seg-only model on this segmentation metric. This indicates that with the right formulation, the segmentation subtask (predicting event boundary) can indeed benefit from joint learning with a related captioning subtask (summarizing event content).

Input formulations Results using SimpleConcat compared to their counterparts using Emb_{TIME} in Table 2, are mixed. While Emb_{TIME} seems to bring non-trivial improvement to models trained from scratch, the training from scratch settings also has the largest variance in our experiments[7]. That said, the Seg+Cap model did achieve its best mIoU score on ViTT using Emb_{TIME}. More work is needed to fully understand the potential of Emb_{TIME}.

Comparison against prior work for YouCook2[8] Table 3 provides a summary of dense video captioning performance on YouCook2 reported in prior work. Some of the prior work (Zhang et al., 2016; Shou et al., 2016; Zhou et al., 2018b) focused only on the segmentation subtask, while some (Zhou et al., 2018c; Shi et al., 2019) approached the end-to-end task as a two-stage task and solved the two subtasks separately. In this context, we find the results from our simple end-to-end sequence generation based approach quite encouraging, and hope this inspires future studies to fully realize the potential of this alternative approach.

Qualitative Analysis We provide a few example model outputs from our Seg+Cap model. More examples can be found in the Appendix A.4.

Figure 5 presents example segmentation results. As reflected in Figure 5(a), a segmentation prediction that is largely correct for a few segments, but is missing out on some ground-truth segments and contains over-segmentation of others can result in a relatively low mIoU score. For examples with relatively high mIoU scores, see Figure 5(b) (taken from the more challenging YouCook2 setting, with gaps between ground-truth segments), and Figure 5(c) (taken from ViTT, with no gaps between segments).

Next, we observe that caption quality is good when the segment boundary prediction is highly accurate: if we restrict to segments with IoU $\geq 90\%$ between the prediction and the target, the average ROUGE-L score for corresponding captions is 30.18 for YouCook2 and 44.33 for ViTT. Table 4 presents qualitative examples.

6 Conclusion

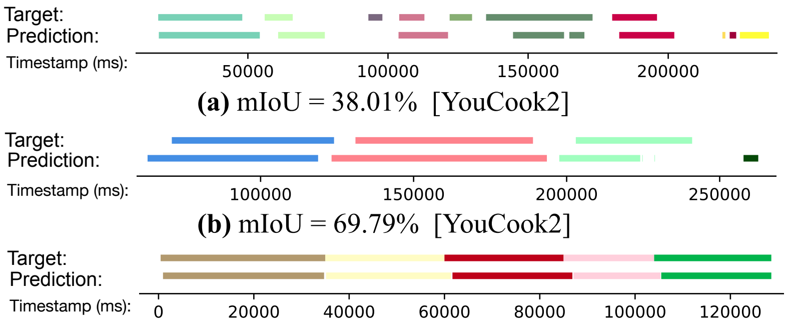
In this paper, we describe different task formulations for solving the dense video captioning task using an end-to-end sequence generation approach, which allows us to leverage pre-trained text-only encoder-decoder models. We conduct experiments on YouCook2 and ViTT in several pretraining settings. Experimental results show that general (T5) and in-domain (WikiHow) text-only pre-trained models both improve video partitioning and segmentation performance, and the gains are cumulative. Also, the segmentation subtask benefits from joint modeling with the captioning subtask. We hope our work can inspire future studies on leveraging pre-trained models, large-scale text corpora and language generation formulations to solve multimodal tasks such as dense video captioning.

Ethical Statement

Our experiments are conducted only on videos with available English ASR annotations, as we inherit this limitation from the available data for this task. We use existing datasets based on public YouTube videos. As a consequence, any videos that are

[7] To the extent that the Seg+Cap model performance in #1 can be considered an outlier: its mIoU scores for the three runs are (11, 11, 26), which resulted in a large std value not seen anywhere else in the table. We looked into these three runs in more details, and our best guess was that one of them incidentally got an advantageous random initialization.

[8] ViTT is a relatively newer dataset and past work has only reported performance of the segment-level captioning subtask using ground-truth segments; we are not aware of existing work reporting end-to-end dense video captioning performance.



(a) mIoU = 38.01% [YouCook2]

(b) mIoU = 69.79% [YouCook2]

(c) mIoU = 91.22% [ViTT]

Figure 5: Example segmentation predictions corresponding to different mIoU scores.

	IoU	**Segment Border (ms)**	**Caption**
Tgt. Pred.	90.0%	[58000.0, 77000.0] [57309.0, 78429.5]	whisk eggs and season with salt whisk the eggs in the deep plate
Tgt. Pred.	99.3%	[28000.0, 45000.0] [28005.0, 44894.0]	chop up the garlic in the food processer chop garlic and place in the food processor
Tgt. Pred.	94.7%	[64199.0, 98080.0] [65710.0, 98380.0]	Preparining remaining ingredients Chopping the remaining ingredients
Tgt. Pred.	97.0%	[65100.0, 124729.0] [63239.5, 124714.5]	Blow-drying the roots Blow-drying hair

Table 4: Example caption predictions where the IoU $\geq 90\%$ between the target (Tgt.) and the predicted (Pred.) segments. The first two examples are from YouCook2, the last two examples are from ViTT.

no longer publicly available on YouTube (e.g., removed by the user) at the time of the study needed to be excluded from our experimental setup.

References

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Apostol Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. Youtube-8m: A large-scale video classification benchmark. *ArXiv*, abs/1609.08675.

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2015. Vqa: Visual question answering. *International Journal of Computer Vision*, 123:4–31.

Prashant Baheti. 2019. Search helps you find key moments in videos.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. 2022. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO Captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. 2021. Sketch, ground, and refine: Top-down dense video captioning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 234–243.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. *ArXiv*, abs/2006.06195.

Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. 2020a. Multimodal pretraining for dense video captioning. In *AACL*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020b. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *ArXiv*, abs/2004.00849.

Vladimir Iashin and Esa Rahtu. 2020. Multi-modal dense video captioning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4117–4126.

Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The kinetics human action video dataset. *ArXiv*, abs/1705.06950.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *ArXiv*, abs/1810.09305.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 706–715.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pretraining. In *AAAI*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly localizing and describing events for dense video captioning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7500.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.

Junyang Lin, An Yang, Yichang Zhang, Jianbin Liu, Jingren Zhou, and Hongxia Yang. 2020. Interbert: Vision-and-language interaction for multi-modal pretraining. *ArXiv*, abs/2003.13198.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *ArXiv*, abs/2002.06353.

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. *ArXiv*, abs/1906.03327.

Jonghwan Mun, L. Yang, Zhou Ren, N. Xu, and Bohyung Han. 2019. Streamlined dense video captioning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6581–6590.

Celie O'Neil-Hart. 2017. Self-directed learning from youtube - think with google.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Di Qi, Lin Su, Jianwei Song, Edward Cui, Taroon Bharti, and Arun Sacheti. 2020. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *ArXiv*, abs/2001.07966.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and M. Zhou. 2019. Dense procedure captioning in narrated instructional videos. In *ACL*.

Zheng Shou, Dongang Wang, and Shih-Fu Chang. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058.

Chen Sun, Austin Myers, Carl Vondrick, Kevin P. Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472.

Hao Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Jingwen Wang, Wenhao Jiang, Lin Ma, W. Liu, and Yong Xu. 2018. Bidirectional attentive fusion with context gating for dense video captioning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7190–7198.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-end dense video captioning with parallel decoding. *ArXiv*, abs/2108.07781.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *ArXiv*, abs/1901.06706.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6724.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. In *ECCV*.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2018b. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

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

A Appendix

A.1 Timestamp markers (T-marker)

Here we describe an alternative way to encode temporal alignment between textual and visual input. Since the frames are extracted at a fixed rate, we can explicitly add time markers to the text input to "mark" out tokens spoken at the corresponding time points. The video features are extracted with a frame rate of 1 frame per second in our work. We insert a time marker for each frame after the last ASR token spoke before the corresponding timestamp. A time marker consists of a special anchor token, followed by the timestamp token (an integer corresponding to the timestamp in seconds).

Performance using this input formuation can be found in the T-marker rows in Table 7. For models trained from scratch, including the timestamp markers can positively impact model performance, indicating that these markers do indeed provide helpful information. However, adding these markers only hurt the performance of any model trained from an existing checkpoint. We hypothesize that this is because the text sequence with frequent markers is too different from the pre-trained datasets, leaving the pre-trained checkpoints less effective for models using this input formulation.

A.2 Ablation studies

Ablation on Input Data Table 5 shows comparisons of different input sources on YouCook2 dense video captioning task. For all three settings (text-only, video-only, text+video), pre-training on WikiHow has the best performance on both subtasks, and using the T5 checkpoint has better performance than training from scratch. With the pre-trained WikiHow checkpoint, the "Text-only" setting has comparable performance as the "Text+Video" setting that takes both the ASR transcript and the video features as input. Using the video features alone results in worse performance, indicating the high value of text transcripts to the captioning task.

Ablation on T5 Checkpoint Table 6 compares performances when using different pre-trained checkpoints on YouCook2 modified dense video captioning. Using either the T5 or the WikiHow T5 checkpoints outperforms the model initialized from random weights, which verifies the effectiveness of pre-training. Since the targets in the end task are markedly different from, say, T5 pretraining targets, we also experimented with loading partial checkpoints (e.g., only encoder weights). Interestingly, using the full checkpoint has better performance than loading only encoder or only decoder weights.

A.3 Comprehensive experimental results

Table 7 provides a more comprehensive summary of our experimental results in the original setting. It is the same experimental setting as Table 2, but we also report additional performance metrics for the segmentation tasks, as well as performance for the T-marker input formulation. Table 8 is again under the same experimental setting, but reports median instead of (mean, std) to summarize the 3 repeats for each setting, so that the metrics are less affected by occasional outliers.

A.4 Qualitative examples

Here we provide more examples for the segmentation subtask and the captioning subtask.

Figure 6 and Figure 7 illustrate several sets of segmentation results predicted by our Seg+Cap model on YouCook2 and ViTT. We can see that the predicted segmentation predictions on ViTT (Figure 7) are relatively more aligned with the ground-truth. This is because ViTT has a comparably simpler formulation with no gaps between video segments. In the more challenging setup on YouCook2 (Figure 6) where the model needs to predict both the start and end point for each segment, the listed examples show that when predictions are mostly correct for a few segments, the IoU scores can be relatively low. Common types of segmentation misalignment include:

- "over-segmentation": the prediction splits a ground-truth span into several sub-chunks (Figure 6 (a)(b)(c));
- "under-segmentation": one predicted segment covers several ground-truth events (Figure 6 (c)(f));
- "prediction-not-covered": the predicted event is not labeled by the ground-truth annotation (Figure 6 (b)(d)(e)).

Table 9 shows examples of the jointly predicted segments and corresponding captions. We have similar findings as in the main paper that when the segment boundary prediction is well aligned with the ground truth, the corresponding captions are often high-quality as well.

Input	Checkpoint	Segmentation			Segmentation + Captioning				
		Precision	Recall	F1	F1	B@4	METEOR	CIDEr	ROUGE-L
Text-only	-	31.86	30.66	31.25	30.39	0.55	1.88	0.07	5.23
	T5	36.22	37.06	36.64	37.89	3.36	4.76	0.28	10.61
	WikiHow T5	**71.13**	**63.77**	**67.25**	**58.71**	9.57	**11.99**	0.85	23.21
Video-only	-	28.02	19.48	22.98	27.5	0.52	1.89	0.07	4.82
	T5	27.43	27.25	27.34	27.86	0.40	1.65	0.05	4.11
	WikiHow T5	25.45	24.93	25.19	23.19	0.42	1.48	0.05	3.84
Text + Video	-	32.53	30.90	31.69	29.09	0.34	1.68	0.06	4.78
	T5	36.96	37.99	37.47	32.58	2.99	4.22	0.26	9.20
	WikiHow T5	71.07	62.76	66.66	57.84	**9.87**	11.96	**0.86**	**23.25**

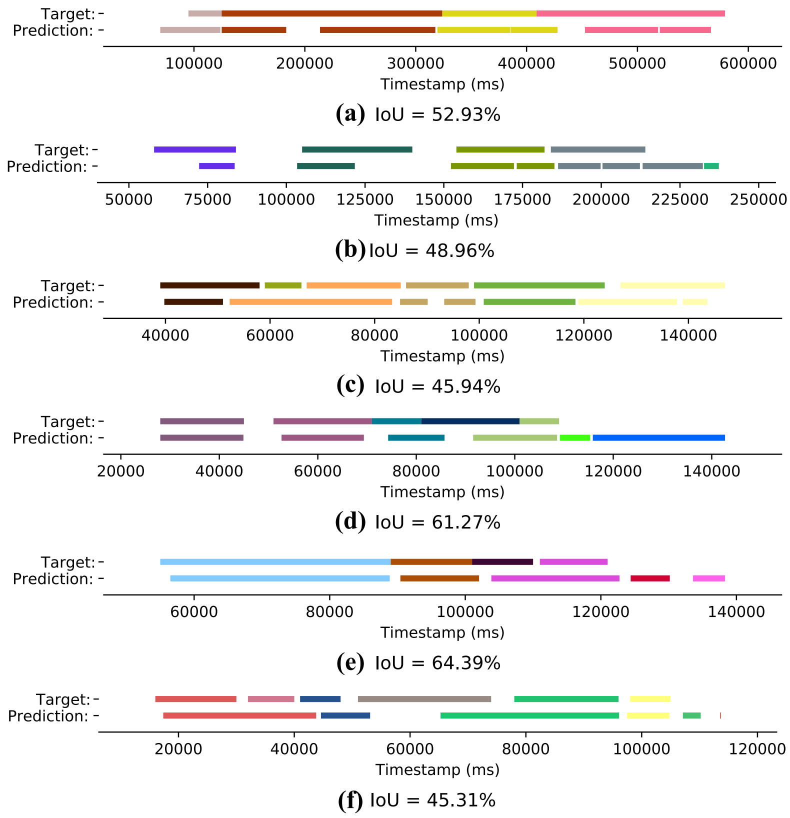
Table 5: Ablation on input modalities. Performance using length-based target formulation on YouCook2 dense video captioning task with IoU threshold=50%. Results are reported on three ablated input settings: “Text-only” feeds in the ASR tokens, “Video-only” reveals the video features, while “Text+Video” provides both the ASR and the video features as input.

Checkpoint	F1	B@4	METEOR	CIDEr	ROUGE-L
-	30.39	0.55	1.88	0.07	5.23
T5 (full)	37.89	3.36	4.76	0.28	10.61
T5 (enc-only)	31.00	0.28	1.93	0.07	4.88
T5 (dec-only)	32.37	1.39	3.02	0.14	7.73
WikiHow T5 (full)	58.71	**9.57**	**11.99**	**0.85**	**23.21**
WikiHow T5 (enc-only)	**59.30**	8.44	11.72	0.80	22.89
WikiHow T5 (dec-only)	36.88	0.99	3.19	0.15	8.00

Table 6: Ablation on pretrained checkpoints. Performance using length-based target formulation on YouCook2 modified dense video captioning with IoU threshold=50%. Results are reported on three settings: “full” loads the complete checkpoint, “enc-only” loads the Transformer encoder weights, while “dec-only” loads the Transformer decoder weights.

*	Input Formulation	Checkpoint?	Seg-only model				Seg+Cap model					
			mIoU	Precision	Recall	F1	mIoU	F1	B@4	METEOR	CIDEr	ROUGE-L
YouCook2												
0	Random Segmentation		20.61 ± 1.04	11.25 ± 0.61	12.49 ± 0.36	10.49 ± 0.59	-	-	-	-	-	-
1	SimpleConcat	-	12.99 ± 1.55	12.24 ± 1.08	8.60 ± 0.90	9.39 ± 0.75	16.45 ± 8.72	11.23 ± 5.16	0.17 ± 0.11	0.66 ± 0.04	0.02 ± 0.01	1.99 ± 0.20
2		T5	24.14 ± 1.07	14.22 ± 0.16	15.09 ± 0.85	14.10 ± 0.44	24.21 ± 1.64	14.20 ± 1.35	0.88 ± 0.04	1.50 ± 0.12	0.09 ± 0.01	3.34 ± 0.27
3		WikiHow	22.58 ± 1.09	13.39 ± 0.96	14.57 ± 1.19	13.27 ± 1.00	23.33 ± 0.79	14.22 ± 0.94	0.67 ± 0.15	1.47 ± 0.05	0.08 ± 0.01	3.51 ± 0.13
4		WikiHow T5	**27.77** ± 0.09	**16.68** ± 1.04	**18.43** ± 0.75	**16.87** ± 0.62	**30.26** ± 1.24	**20.24** ± 1.06	**2.96** ± 0.28	**3.49** ± 0.30	**0.25** ± 0.03	**7.00** ± 0.42
5	+ T-marker	-	20.13 ± 2.59	13.68 ± 1.78	12.18 ± 1.88	12.01 ± 1.91	18.41 ± 2.65	9.99 ± 1.55	0.08 ± 0.02	0.44 ± 0.04	0.01 ± 0.00	1.33 ± 0.15
6		T5	20.29 ± 1.30	12.13 ± 2.52	11.43 ± 0.64	11.09 ± 1.38	22.12 ± 1.29	12.56 ± 0.74	0.88 ± 0.23	1.38 ± 0.22	0.08 ± 0.02	3.07 ± 0.39
7		WikiHow	19.98 ± 0.55	10.54 ± 1.36	12.11 ± 1.29	10.68 ± 1.32	20.84 ± 1.02	11.82 ± 0.64	0.39 ± 0.05	0.99 ± 0.09	0.05 ± 0.00	2.44 ± 0.18
8		WikiHow T5	20.98 ± 0.69	11.99 ± 1.07	12.49 ± 0.60	11.86 ± 0.82	20.22 ± 0.70	11.20 ± 0.70	0.38 ± 0.08	0.92 ± 0.05	0.05 ± 0.00	2.27 ± 0.13
9	+ Emb$_{\text{TIME}}$	-	18.51 ± 1.95	10.85 ± 0.59	11.42 ± 1.16	10.29 ± 0.58	18.71 ± 0.17	9.80 ± 0.80	0.12 ± 0.07	0.48 ± 0.08	0.02 ± 0.01	1.41 ± 0.22
10		T5	23.02 ± 1.05	13.52 ± 0.76	14.15 ± 0.94	13.23 ± 0.77	23.96 ± 0.08	15.44 ± 0.67	1.32 ± 0.08	1.91 ± 0.07	0.11 ± 0.01	4.20 ± 0.13
11		WikiHow	21.68 ± 1.93	13.13 ± 1.42	13.88 ± 1.60	12.83 ± 1.41	21.88 ± 0.86	13.15 ± 0.74	0.69 ± 0.19	1.30 ± 0.07	0.07 ± 0.01	3.06 ± 0.13
12		WikiHow T5	26.51 ± 0.45	15.61 ± 0.61	17.08 ± 0.58	15.82 ± 0.62	28.70 ± 0.92	18.71 ± 0.94	2.58 ± 0.19	3.23 ± 0.10	0.22 ± 0.01	6.45 ± 0.17
ViTT												
13	Random Segmentation		21.90 ± 0.15	12.22 ± 0.09	16.12 ± 0.25	12.48 ± 0.10	-	-	-	-	-	-
14	SimpleConcat	-	33.85 ± 0.70	23.54 ± 0.36	24.04 ± 0.40	22.98 ± 0.22	32.69 ± 0.71	22.49 ± 0.36	0.11 ± 0.01	3.76 ± 0.35	0.08 ± 0.01	3.86 ± 0.28
15		T5	37.89 ± 0.10	28.16 ± 1.18	27.15 ± 0.19	27.15 ± 0.53	38.07 ± 0.65	27.39 ± 0.91	0.57 ± 0.03	5.92 ± 0.37	0.16 ± 0.02	6.59 ± 0.69
16		WikiHow	38.20 ± 0.27	26.95 ± 0.67	27.71 ± 0.25	26.85 ± 0.41	37.80 ± 0.62	26.74 ± 0.81	0.40 ± 0.07	5.48 ± 0.18	0.14 ± 0.01	6.02 ± 0.34
17		WikiHow T5	**41.87** ± 0.26	**31.75** ± 1.94	**31.74** ± 0.34	**31.26** ± 1.10	42.40 ± 0.30	32.01 ± 0.50	**1.29** ± 0.07	**8.10** ± 0.34	**0.25** ± 0.01	**9.26** ± 0.39
18	+ T-marker	-	32.19 ± 1.17	20.05 ± 1.89	21.62 ± 0.82	20.04 ± 0.48	32.03 ± 0.14	20.89 ± 0.28	0.05 ± 0.00	2.96 ± 0.13	0.06 ± 0.00	2.93 ± 0.07
19		T5	34.94 ± 0.37	21.24 ± 0.11	23.95 ± 0.41	22.07 ± 0.21	37.56 ± 0.78	27.50 ± 0.69	0.59 ± 0.09	5.11 ± 0.52	0.16 ± 0.01	6.26 ± 0.56
20		WikiHow	33.00 ± 0.10	19.13 ± 0.87	22.02 ± 0.13	20.05 ± 0.54	35.14 ± 0.99	22.88 ± 0.41	0.23 ± 0.04	3.51 ± 0.14	0.09 ± 0.01	4.12 ± 0.37
21		WikiHow T5	34.23 ± 0.55	21.01 ± 1.34	23.26 ± 0.51	21.62 ± 0.94	33.20 ± 1.65	19.63 ± 0.98	0.16 ± 0.02	3.01 ± 0.22	0.08 ± 0.01	3.40 ± 0.36
22	+ Emb$_{\text{TIME}}$	-	33.89 ± 0.21	20.75 ± 2.37	23.69 ± 0.08	21.27 ± 1.49	35.37 ± 3.18	22.28 ± 0.49	0.04 ± 0.03	3.42 ± 0.61	0.07 ± 0.01	3.28 ± 0.83
23		T5	37.78 ± 0.15	25.98 ± 0.20	27.12 ± 0.16	26.05 ± 0.16	38.50 ± 0.55	27.95 ± 0.46	0.75 ± 0.10	6.37 ± 0.39	0.18 ± 0.01	7.19 ± 0.48
24		WikiHow	37.27 ± 0.08	25.96 ± 0.38	26.87 ± 0.04	25.91 ± 0.21	36.97 ± 0.48	26.37 ± 0.36	0.38 ± 0.06	5.31 ± 0.06	0.13 ± 0.01	5.82 ± 0.23
25		WikiHow T5	41.64 ± 0.12	31.07 ± 0.67	31.53 ± 0.12	30.84 ± 0.33	**43.22** ± 0.72	**32.49** ± **0.25**	1.22 ± 0.08	8.05 ± 0.20	**0.25** ± 0.01	9.18 ± 0.45

Table 7: Dense video captioning performance on YouCook2 and ViTT test sets with the length-based and the Timestamp markers formulations. We report the evaluation results (mean ± std) with models initialized from random weights, T5 checkpoints, WikiHow checkpoints, and T5 checkpoints further pretrained on WikiHow.



(a) IoU = 52.93%

(b) IoU = 48.96%

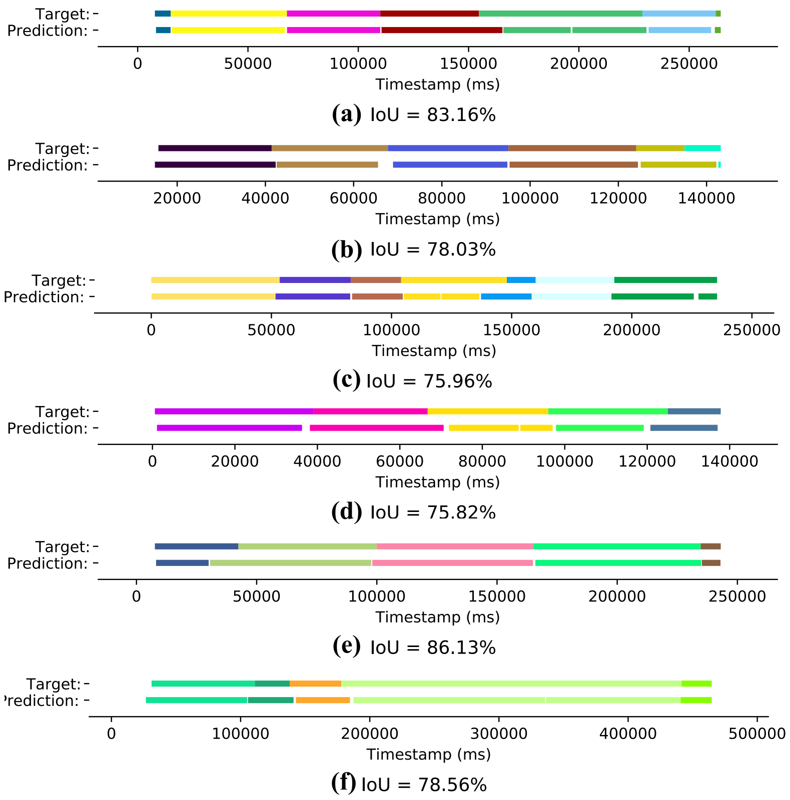
(c) IoU = 45.94%

(d) IoU = 61.27%

(e) IoU = 64.39%

(f) IoU = 45.31%

Figure 6: Examples of our Seg+Cap model's segmentation performance on YouCook2.



(a) IoU = 83.16%

(b) IoU = 78.03%

(c) IoU = 75.96%

(d) IoU = 75.82%

(e) IoU = 86.13%

(f) IoU = 78.56%

Figure 7: Examples of our Seg+Cap model's segmentation performance on ViTT.

*	Dataset	Input Formulation	Checkpoint	Seg-only model				Seg+Cap model					
				mIoU	Precision	Recall	F1	mIoU	F1	B@4	METEOR	CIDEr	ROUGE-L
0	YouCook2	Random Segmentation		20.10	11.14	12.53	10.69	-	-	-	-	-	-
1		SimpleConcat	-	12.83	12.80	8.62	9.07	11.47	8.78	0.22	0.64	0.03	1.95
2			T5	24.18	14.19	14.83	13.89	25.13	14.09	0.86	1.47	0.09	3.39
3			WikiHow	22.36	13.11	14.42	12.99	23.00	14.16	0.66	1.50	0.08	3.48
4			WikiHow T5	**27.81**	**17.21**	**18.16**	**17.01**	**30.97**	**20.57**	**2.85**	**3.48**	**0.24**	**7.02**
5		+ T-marker	-	18.82	14.42	11.28	11.38	17.18	9.53	0.06	0.45	0.01	1.34
6			T5	20.91	13.21	11.48	11.65	22.75	12.87	0.90	1.42	0.08	3.19
7			WikiHow	19.76	10.17	11.52	10.19	21.09	11.79	0.37	0.94	0.05	2.35
8			WikiHow T5	21.26	12.34	12.83	12.24	20.56	11.35	0.38	0.89	0.05	2.31
9		+ Emb$_{\text{TIME}}$	-	19.52	10.99	11.70	10.26	18.77	9.83	0.11	0.52	0.02	1.46
10			T5	22.93	13.84	13.78	13.30	24.00	15.70	1.34	1.90	0.11	4.24
11			WikiHow	21.41	13.11	13.88	12.76	22.08	13.18	0.79	1.30	0.07	3.09
12			WikiHow T5	26.61	15.86	17.28	16.08	28.80	18.41	2.67	3.18	0.23	6.41
13	ViTT	Random Segmentation		21.93	12.22	16.07	12.41	-	-	-	-	-	-
14		SimpleConcat	-	33.74	23.71	23.95	23.10	33.10	22.59	0.12	3.78	0.08	3.87
15			T5	37.90	28.28	27.14	27.13	38.35	27.66	0.57	5.85	0.15	6.36
16			WikiHow	38.23	26.82	27.78	26.89	37.75	26.92	0.44	5.58	0.14	6.06
17			WikiHow T5	**41.78**	**31.00**	**31.62**	**30.78**	42.25	31.85	**1.34**	7.97	**0.25**	**9.21**
18		+ T-marker	-	32.62	19.94	22.02	20.27	32.01	20.90	0.05	2.96	0.06	2.95
19			T5	34.83	21.20	23.89	22.08	37.36	27.80	0.57	5.31	0.16	6.46
20			WikiHow	33.00	19.12	22.09	20.09	35.54	23.09	0.23	3.43	0.09	4.03
21			WikiHow T5	33.92	21.12	23.01	21.52	34.04	19.46	0.16	2.96	0.07	3.23
22		+ Emb$_{\text{TIME}}$	-	33.79	21.21	23.68	21.61	34.56	22.37	0.05	3.12	0.06	2.92
23			T5	37.75	25.97	27.13	26.13	38.44	27.94	0.69	6.18	0.18	7.15
24			WikiHow	37.22	25.85	26.86	25.84	37.07	26.39	0.37	5.28	0.13	5.73
25			WikiHow T5	41.62	30.76	31.52	30.68	**43.51**	**32.50**	1.19	**8.05**	**0.25**	9.02

Table 8: Performance on the dense video captioning on YouCook2 and ViTT test set with the length-based and the Timestamp markers formulations. We report the evaluation results with models initialized from random weights, T5 checkpoints, WikiHow checkpoints, and T5 checkpoints further pretrained on WikiHow. We ran 3 sets of repeating experiments for each setting, and report the **median** value on each metric in this Table.

Dataset		IoU	Segment Border (ms)	Caption
Youcook2	Tgt.	82.2%	[49000.0, 67000.0]	chop 2 garlic cloves grate ginger about 2 tsp and green onions finely
	Pred.		[47114.0, 65354.5]	chop some garlic ginger and green onions and put them in a bowl
	Tgt.	82.4%	[73000.0, 117000.0]	mix an egg milk and the mashed potatoes
	Pred.		[72194.0, 109904.5]	mix the egg and milk with the potato
	Tgt.	81.1%	[89000.0, 101000.0]	mix and boil the ingredients
	Pred.		[90424.5, 102034.0]	add miso paste soy sauce diced vegetables and mushrooms to boiling water
	Tgt.	82.1%	[18000.0, 48000.0]	heat butter in a pan and cook bacon in it
	Pred.		[18224.0, 54254.0]	fry pancetta in a pan with bacon
ViTT	Tgt.	90.5%	[147890.0, 189680.0]	Dipping sticks then cake balls
	Pred.		[148905.0, 192934.0]	Dipping cake balls in candy melts
	Tgt.	90.2%	[209050.0, 253460.0]	Buttering in between baking, baking continues
	Pred.		[211445.0, 255634.5]	Brushing the dough with butter
	Tgt.	90.6%	[209630.0, 272000.0]	Stretching the hamstrings
	Pred.		[213230.0, 269750.0]	Performing the hamstring stretch
	Tgt.	92.3%	[104000.0, 144000.0]	Adding more layers
	Pred.		[105404.0, 145815.0]	Repeating the same process

Table 9: Examples of the jointly predicted segments and corresponding captions for YouCook2 and ViTT generated by our Seg+Cap model. Tgt.: Target. Pred.: Prediction.