Vid2Seq: Large-Scale Pretraining of a Visual Language Model for Dense Video Captioning - Supplementary Material

Antoine Yang[†*] Arsha Nagrani[§] Paul Hongsuck Seo[§] Antoine Miech[♯]
Jordi Pont-Tuset[§] Ivan Laptev[†] Josef Sivic[¶] Cordelia Schmid[§]

[§]Google Research [†]Inria Paris and Département d'informatique de l'ENS, CNRS, PSL Research University
[♯] DeepMind [¶]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague

https://antoyang.github.io/vid2seq.html

Overview

In this Supplementary Material, we present the following additional material:

(i) Additional qualitative examples of dense video captioning predictions (Section A).

(ii) Additional information about our experimental setup (Section B);

(iii) Additional experimental results (Section C), including an ablation on the importance of pretraining for few-shot dense video captioning (Section C.1) and additional ablation studies in the standard fully-supervised dense video captioning setting (Section C.2).

A. Qualitative examples of dense video captioning predictions

In Figure 4 of the main paper, we show qualitative results of dense event captioning by our Vid2Seq model. Here in Figures 1 and 2 , we show additional results on examples from the YouCook2 and ActivityNet Captions datasets. These results show that Vid2Seq can predict meaningful dense captions and event boundaries in diverse scenarios, with or without transcribed speech input, *e.g.* series of instructions in cooking recipes (Figure 1) or actions in human sports or leisure activities (first three examples in Figure 2). The last example in Figure 2 illustrates a failure case where the model hallucinates events that are not visually grounded such as 'one man hats off to the camera'.

B. Experimental setup

In this section, we complement the information provided in Section 4.1 of the main paper about the datasets we use (Section B.1). We also give additional implementation details (Section B.2).

[*]This work was done when the first author was an intern at Google.

B.1. Datasets

YT-Temporal-1B [16] consists of 18.821M unlabeled narrated videos covering about 150 years of video content for pretraining. Compared with HowTo100M [8], this dataset was created to cover a wider range of domains and not only instructional videos.

HowTo100M [8] consists of 1.221M unlabeled narrated instructional videos covering about 15 years of video content for pretraining.

YouCook2 [18] has 1,790 untrimmed videos of cooking procedures. On average, each video lasts 320s and is annotated with 7.7 temporally-localized imperative sentences. The dataset is split into 1,333 videos for training and 457 videos for validation.

ViTT [3] consists of 7,672 untrimmed instructional videos from the YouTube-8M dataset [1]. Compared to YouCook2, ViTT was created to better reflect the distribution of instructional videos in the wild. On average, each video lasts 250s and is annotated with 7.1 temporally-localized short tags. The dataset is split into 5,476, 1,102 and 1,094 videos for training, validation and testing, respectively. Videos in the validation and test sets are provided with multiple sets of dense event captioning annotations. Following [3], we treat each set of annotations as a single example during evaluation and discard videos with more than 3 sets of annotations.

ActivityNet-Captions [5] contains 14,934 untrimmed videos of various human activities. Different from YouCook2 and ViTT where most videos contain transcribed speech content, we find that 68% of videos in ActivityNet Captions do not have transcribed narration. On average, each video lasts 120s and is annotated with 3.7 temporally-localized sentences. The dataset is split into 10,009 and 4,925 videos for training and validation, respectively. Videos in the validation set are provided with two sets of dense video captioning annotations. Following prior work [14], we use both sets of annotations for evaluation,

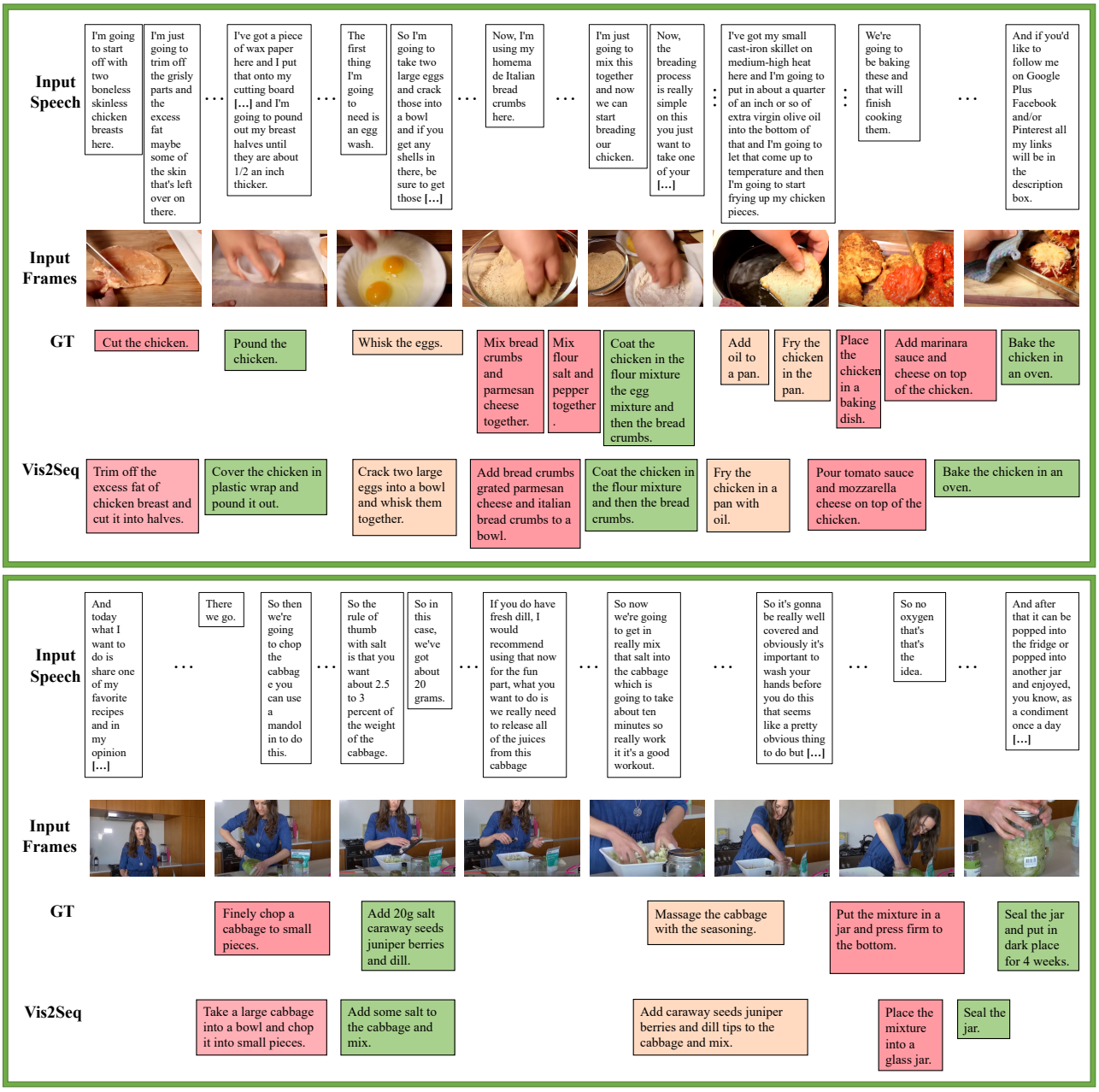

Figure 1. Examples of dense event captioning predictions of Vid2Seq on the validation set of YouCook2, compared with ground-truth.

by computing the average of the scores over each set for SODA_c and by using the standard evaluation tool [5] for all other dense event captioning metrics. For video paragraph captioning, we follow [14] and report results on the 'val-ae' split that includes 2,460 videos [6, 17].

MSR-VTT [15] consists of 10,000 open domain video clips. The duration of each video clip is between 10 and 30 seconds. 20 natural language descriptions are manually annotated for each clip. The dataset is split into 6,513, 497 and 2,990 videos for training, validation and testing, respectively.

MSVD [2] consists of 1,970 open domain video clips. The duration of each video clip is between 10 and 30 seconds. Each video clip has roughly 40 manually annotated captions. The dataset is split into 1,200, 100 and 670 videos for training, validation and testing, respectively.

B.2. Implementation details

Architecture. The visual temporal transformer encoder f^t, the text encoder g^t and the text decoder h^t all have

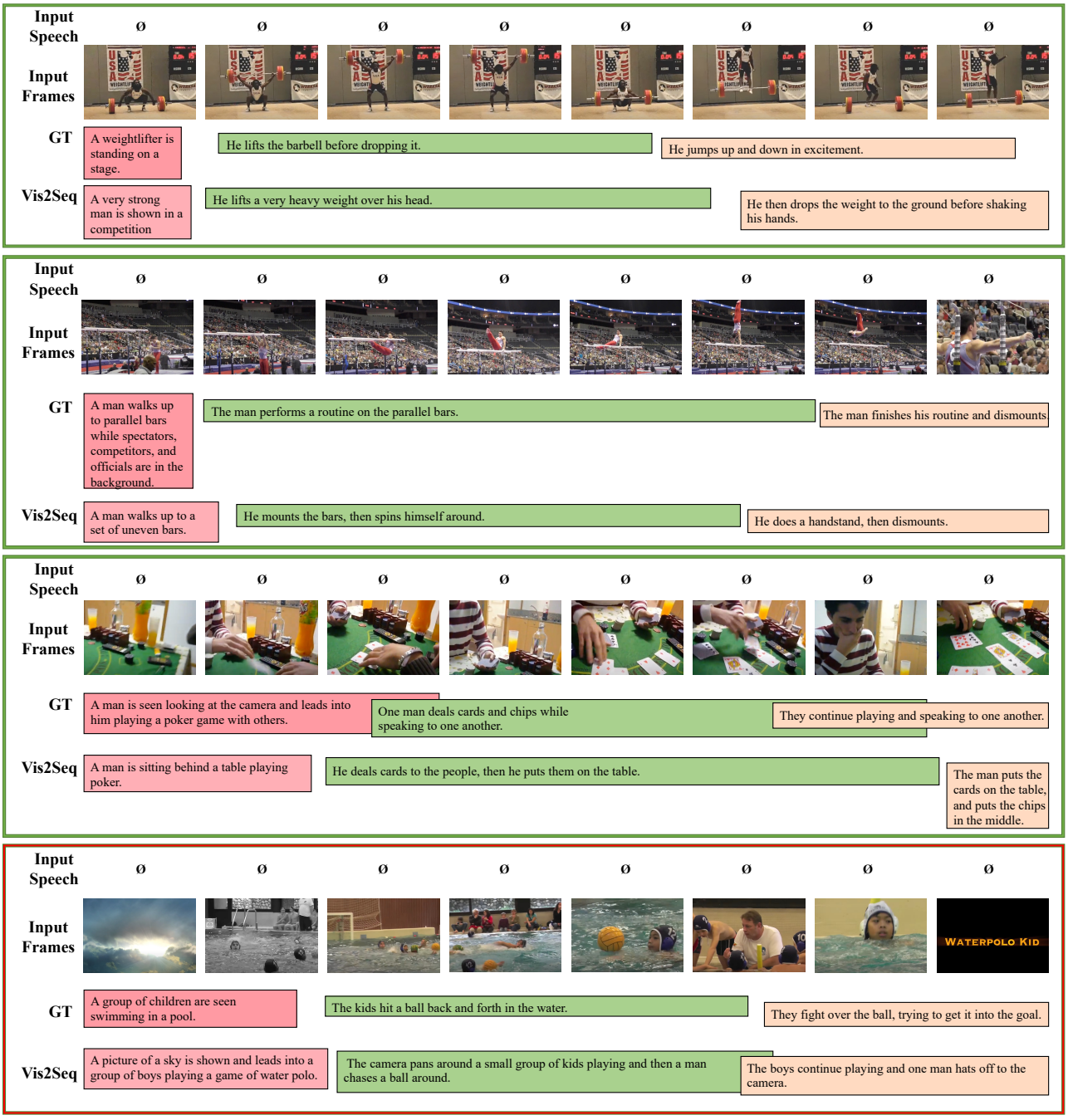

Figure 2. Examples of dense event captioning predictions by Vid2Seq on the validation set of ActivityNet Captions, compared with ground-truth. The first three examples show successful predictions, while the last example illustrates a failure case where the model hallucinates events that are not visually grounded ('one man hats off to the camera'). Note that in all of these videos, there is no transcribed speech.

12 layers, 12 heads, embedding dimension 768, and MLP hidden dimension of 2048. The text encoder and decoder sequences are truncated or padded to $L = S = 1000$ tokens during pretraining, and $S = 1000$ and $L = 256$ tokens during finetuning. At inference, we use beam search decoding where we track the top 4 sequences and apply a length normalization of 0.6.

Training. We use the Adam optimizer [4] with $\beta = (0.9, 0.999)$ and no weight decay. During pretraining, we use a learning rate of $1e^{-4}$, warming it up linearly (from 0) for the first 1000 iterations, and keeping it constant for

	Data	Pretrain	YouCook2			ViTT			ActivityNet		
			S	C	M	S	C	M	S	C	M
1.	1%	✗	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
2.	1%	✓	2.4	10.1	3.3	2.0	7.4	1.9	2.2	6.2	3.2
3.	10%	✗	0.1	0.0	0.2	3.3	0.4	3.3	3.4	11.9	4.6
4.	10%	✓	3.8	18.4	5.2	10.7	28.6	6.0	4.3	20.0	6.1
5.	50%	✗	1.8	8.5	2.4	6.5	18.7	3.9	4.6	13.1	6.3
6.	50%	✓	6.2	32.1	7.6	12.5	38.8	7.8	5.4	27.5	7.8
7.	100%	✗	4.0	18.0	4.6	7.9	21.2	6.2	5.4	18.8	7.1
8.	100%	✓	**7.9**	**47.1**	**9.3**	**13.5**	**43.5**	**8.5**	**5.8**	**30.1**	**8.5**

Table 1. **Impact of our pretraining on few-shot dense event captioning**, by finetuning Vid2Seq using a small fraction of the downstream training dataset.

the remaining iterations. During finetuning, we use a learning rate of $3e{-}4$, warming it up linearly (from 0) for the first 10% of iterations, followed by a cosine decay (down to 0) for the remaining 90%. During finetuning, we use a batch size of 32 videos split on 16 TPU v4 chips. We finetune for 40 epochs on YouCook2, 20 epochs on ActivityNet Captions and ViTT, 5 epochs on MSR-VTT and 10 epochs on MSVD. We clip the maximum norm of the gradient to 0.1 during pretraining, and 1 during finetuning. For data augmentation, we use random temporal cropping. For regularization, we use label smoothing [13] with value 0.1 and dropout [11] with probability 0.1.

C. Experiments

In this section, we provide additional experiments that complement the results presented in Section 4 of the main paper. We first show the importance of pretraining in our proposed few-shot setting in Section C.1. Then we provide additional ablation studies in the standard fully-supervised setting in Section C.2, where we ablate various factors including pretraining on long narrated videos, the pretraining dataset and the size of the visual backbone, the time tokenization process and the number of time tokens, the sequence construction process, the temporal positional embeddings and the initialization of the language model.

C.1. Importance of pretraining in few-shot settings

In Section 4.2 of the main paper, we show the benefits of our pretraining method in the fully-supervised setting, *i.e.* when using 100% of the downstream training dataset. In Table 1, we further show that our pretraining method has a considerable importance in the few-shot setting defined in Section 4.4 of the main paper, *i.e.* when using a smaller fraction of the downstream training dataset. In particular, our pretraining method enables our Vid2Seq model to have a non zero performance when using only 1% of the downstream training dataset (rows 1 and 2).

	Max number of narrations	YouCook2			ActivityNet		
		S	C	F1	S	C	F1
1.	*No pretraining*	4.0	18.0	18.1	5.4	18.8	49.2
2.	1	6.0	32.1	22.1	5.1	22.9	48.1
3.	10	6.5	34.6	23.6	5.4	27.1	50.3
4.	∞	**7.9**	**47.1**	**27.3**	**5.8**	**30.1**	**52.4**

Table 2. **Ablation showing the importance of pretraining on long narrated videos**, by varying the maximum number of narration sentences that a randomly cropped video can cover. ∞ means the cropping is unrestricted and can sample arbitrarily long videos.

	Pretraining Data	Model	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	ImageNet	ViT-B/16	6.6	40.2	24.3	4.5	17.2	49.3
2.	CLIP	ViT-B/16	7.7	46.3	26.5	5.6	28.4	51.7
3.	CLIP	ViT-L/14	**7.9**	**47.1**	**27.3**	**5.8**	**30.1**	**52.4**

Table 3. **Ablation on the pretraining data and model size of the visual backbone f^s.**

C.2. Additional ablation studies

We here complement ablation studies reported in Section 4.2 of the main paper, using the same default settings, evaluation metrics and downstream datasets.

Pretraining on long narrated videos. In Table 1 of the main paper, we show the benefits of pretraining on untrimmed videos in comparison with the standard practice of pretraining on short, trimmed, video-speech segments [3, 7, 10]. In Table 2, we further evaluate the importance of sampling long narrated videos during pretraining. By default, at each training iteration, we randomly temporally crop each narrated video without constraints, resulting in a video that can span over hundreds of transcribed speech sentences. We here evaluate a baseline that constrains this cropping process such that the cropped video only spans over a given maximum number of narration sentences. Even with a maximum of 10 narration sentences, this baseline significantly underperforms our model trained in default settings where we sample longer untrimmed narrated videos (rows 1, 2 and 3). This demonstrates that our model benefits from pretraining on long narrated videos.

Visual features. In Table 4 of the main paper, we show the benefits of scaling up the size of the pretraining dataset of narrated videos and the size of the language model. In Table 3, we further analyze the importance of the pretraining dataset and size of the visual backbone f^s. We find that CLIP pretraining [9] considerably improves over ImageNet pretraining [12] with the same ViT-B/16 visual backbone model (row 2 vs 1). Furthermore, scaling up the visual backbone size from ViT-B/16 to ViT-L/14 brings additional improvements (row 3 vs 2).

	Tokenization	N	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	Absolute	20	0.3	0.2	0.9	3.2	23.0	23.1
2.	Absolute	100	3.5	25.7	12.0	4.8	25.5	41.5
3.	Absolute	500	**7.9**	39.8	24.3	5.4	28.1	48.6
4.	Relative	20	7.2	39.6	23.7	5.6	29.0	49.4
5.	Relative	100	**7.9**	**47.1**	**27.3**	**5.8**	**30.1**	52.4
6.	Relative	500	7.2	40.0	25.0	5.7	28.6	**52.5**

Table 4. **Ablation on time tokenization (relative or absolute) and the number of time tokens N.**

	Dot symbol between segments	Time tokens Position	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	✗	*After text*	7.9	48.3	26.7	5.6	29.8	51.1
2.	✓	*After text*	**8.3**	**50.9**	26.2	5.7	**30.4**	51.8
3.	✗	*Before text*	8.0	50.0	**27.3**	5.6	28.2	50.7
4.	✓	*Before text*	7.9	47.1	**27.3**	**5.8**	30.1	**52.4**

Table 5. **Ablation on the sequence construction process.**

Time tokenization and number of time tokens. In Table 4, we further ablate the time tokenization process presented in Section 3.1 of the main paper. Our default time tokens represent relative timestamps in a video, as we quantize a video of duration T into N equally-spaced timestamps. Another possibility is to use time tokens that represent absolute timestamps in the video, *i.e.* the k-th token represents the k-th second in the video. For both these variants, we vary the number of time tokens N. For the relative time tokens, increasing N makes the quantization more fine-grained but also spreads the data into more time tokens. On the other hand, for the absolute time tokens, increasing N increases the video duration that the time tokens can cover. We find that the best dense video captioning results are obtained with the relative time tokens and $N = 100$ time tokens (row 5).

Sequence construction. In Table 5, we further ablate the sequence construction process presented in Section 3.1 of the main paper. Our default sequence inserts the start and end time tokens of each segment before its corresponding text sentence. Another possibility is to insert time tokens after each corresponding text sentence. We find that both variants achieve similar results (rows 2 and 4), with the default sequence (row 4) resulting in slightly higher event localization performance (F1 Score) but slightly lower dense captioning results overall. Furthermore, we observe that the dot symbols indicating the separation between different events have low importance (rows 1 and 2, rows 3 and 4).

Temporal positional embeddings. In Table 1 of the main paper, we show that time tokens in the speech sequence provide temporal information about the speech transcript to our model. In Table 6, we also evaluate the importance of the temporal positional embeddings which communicate temporal information from the visual stream to our model. We find that these temporal embeddings are beneficial (row 2 vs 1).

	Temporal embeddings	YouCook2			ActivityNet		
		S	C	F1	S	C	F1
1.	✗	6.8	42.0	24.9	5.3	27.0	50.6
2.	✓	**7.9**	**47.1**	**27.3**	**5.8**	**30.1**	**52.4**

Table 6. **Ablation on the temporal positional embeddings.**

	Language Model Initialization	Video-text Pretraining	YouCook2			ActivityNet		
			S	C	F1	S	C	F1
1.	✗	✗	0.9	4.2	7.6	4.3	23.7	41.2
2.	✓	✗	4.0	18.0	18.1	5.4	18.8	49.2
3.	✗	✓	**8.8**	**51.3**	**28.4**	5.7	28.7	51.2
4.	✓	✓	7.9	47.1	27.3	**5.8**	**30.1**	**52.4**

Table 7. **Ablation on language model initialization and pretraining.**

Language model initialization and pretraining. In Table 4 of the main paper, we show the benefits of using T5-Base instead of T5-Small. In Table 7, we further investigate the importance of initializing the language model from weights pretrained on Web text. Without pretraining on narrated videos, we find that text-only initialization is helpful (rows 1 and 2). Interestingly, after pretraining on narrated videos, we find that text-only initialization has little importance (rows 3 and 4), as it slightly improves the performance on ActivityNet Captions while resulting in a slight drop of performance on YouCook2. We believe that this may be because of the domain gap between Web text and the imperative-style dense captions in YouCook2, which are more similar to transcribed speech in YT-Temporal-1B.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1

[2] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 2

[3] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. In *AACL-IJCNLP*, 2020. 1, 4

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[5] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 1, 2

[6] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara L Berg, and Mohit Bansal. MART: Memory-augmented recurrent transformer for coherent video paragraph captioning. In *ACL*, 2020. 2

[7] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. UniViLM: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 4

[8] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 1

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4

[10] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *CVPR*, 2022. 4

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014. 4

[12] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. In *TMLR*, 2022. 4

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 4

[14] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *ICCV*, 2021. 1, 2

[15] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2

[16] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. MERLOT Reserve: Neural script knowledge through vision and language and sound. In *CVPR*, 2022. 1

[17] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *CVPR*, 2019. 2

[18] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 1