HowToCaption: Prompting LLMs to Transform Video Annotations at Scale

Nina Shvetsova*[1,2,3], Anna Kukleva*[2], Xudong Hong[2,6], Christian Rupprecht[4], Bernt Schiele[2], Hilde Kuehne[1,3,5]

[1]Goethe University Frankfurt, [2]MPI for Informatics, SIC, [3]University of Bonn, [4]University of Oxford, [5]MIT-IBM Watson AI Lab, [6]Saarland University
{nshvetso,akukleva}@mpi-inf.mpg.de

Abstract. Instructional videos are a common source for learning text-video or even multimodal representations by leveraging subtitles extracted with automatic speech recognition systems (ASR) from the audio signal in the videos. However, in contrast to human-annotated captions, both speech and subtitles naturally differ from the visual content of the videos and thus provide only noisy supervision. As a result, large-scale annotation-free web video training data remains sub-optimal for training text-video models. In this work, we propose to leverage the capabilities of large language models (LLMs) to obtain high-quality video descriptions aligned with videos at scale. Specifically, we prompt an LLM to create plausible video captions based on ASR subtitles of instructional videos. To this end, we introduce a prompting method that is able to take into account a longer text of subtitles, allowing us to capture the contextual information beyond one single sentence. We further prompt the LLM to generate timestamps for each produced caption based on the timestamps of the subtitles and finally align the generated captions to the video temporally. In this way, we obtain human-style video captions at scale without human supervision. We apply our method to the subtitles of the HowTo100M dataset, creating a new large-scale dataset, HowToCaption. Our evaluation shows that the resulting captions not only significantly improve the performance over many different benchmark datasets for zero-shot text-video retrieval and video captioning, but also lead to a disentangling of textual narration from the audio, boosting the performance in text-video-audio tasks.[1]

Keywords: Video-Language Dataset · LLM · Instructional Videos

1 Introduction

Textual descriptions of visual information allow for navigating large amounts of visual data. Recently, image-text cross-modal learning has achieved remarkable performance in many downstream tasks by pre-training on large-scale web

* Equal contribution.
[1] All data and code is available at https://github.com/ninatu/howtocaption.

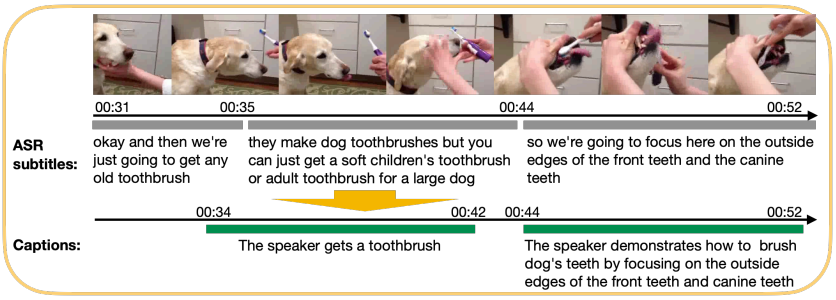


Fig. 1: ASR subtitles deviate from human-written captions: they contain a lot of filler phrases, e.g., "we're going to", and extra information, e.g., "they make dog toothbrushes". We propose to generate human-style video captions based on the ASR subtitles and their timestamps that we further temporally realign with the video.

datasets consisting of text-image pairs [18, 39, 45]. To collect video data on a similar scale, media platforms such as YouTube can be used as a great source of freely available videos [1, 33, 48, 59, 65, 67]. Most of these videos include some narrations, *e.g.*, in instructional videos [33], people explain and show how to accomplish one or another task. To transform spoken language from the videos into subtitles, current automatic speech recognition (ASR) systems [40] can be used, providing aligned text-video annotated pairs for free. This automatic supervisory signal can easily scale to large video datasets. However, such video web data poses additional challenges [16, 33]: (1) spoken and visual information in the video can deviate from each other, *e.g.*, when speakers provide information beyond what is visible or when spoken instructions do not temporally align with the actions shown, (2) speech contains filler words and phrases, such as "I'm going to", and can be incomplete and contain grammatical errors, and (3) ASR transcripts usually do not have punctuation and may contain errors (Fig. 1). Therefore, ASR subtitles provide only weak, noisy supervision for videos.

To address this problem, we propose a new framework, HowToCaption, that leverages large language models (LLMs) [10] to generate human-style captions on a large scale for web-video instructional datasets based on corresponding ASR subtitles (Fig. 1). By carefully designing prompts, we show that the LLM can effectively map long, noisy subtitles into concise and descriptive human-style video captions. Moreover, we obtain an initial temporal alignment of the generated captions to the video based on the ASR timestamps by tasking the LLM to predict timestamps for each caption. For additional quality improvement, we apply alignment and filtering within short temporal windows with respect to the predicted timestamp. This approach can generate aligned text-video pairs on a large scale without any human intervention.

Beyond providing better annotation, the new captions provide the advantage that they are no longer a direct output of the speech signal, thus effectively decoupling audio and text. Current methods usually avoid using audio [16, 32],

as the ASR subtitles are directly derived from the speech, thus leading to the problem that any text-to-audio+video retrieval would mainly retrieve the closest speech signal while disregarding the video [34,47]. Being able to generate captions that deviate from the speech thus allows to extend retrieval to audio+video without the need for fine-tuned regularization, as used in [47].

To verify the effectiveness of the proposed HowToCaption method, we generate new captions for the large-scale HowTo100M dataset [33], obtaining a new *HowToCaption* dataset. We evaluate the quality of the improved textual descriptions on various challenging downstream tasks over four different datasets, namely YouCook2 [67], MSR-VTT [58], MSVD [8], and LSMDC [43]. It shows that the generated captions not only provide a better training signal but also allow for a decoupling of speech and caption annotation, allowing a retrieval based on audio, vision, and subtitles at scale. We release the new HowToCaption dataset with high-quality textual descriptions to show the potential of generated captions for web text-video pairs. We summarize the contributions of the paper as follows:

- We propose a HowToCaption method to efficiently convert noisy ASR subtitles of instructional videos into accurate video captions, which leverages recent advances in LLMs and generates high-quality video captions at scale without any human supervision.
- We create a new HowToCaption dataset with high-quality human-style textual descriptions with our proposed HowToCaption method.
- Utilizing the HowToCaption dataset for training text-video models allows us to significantly improve the performance over many benchmarks for text-to-video retrieval and video captioning. Moreover, since the new textual annotation allows us to disentangle audio and language modalities in the instructional videos, where the ASR subtitles were highly correlated to audio, we show a boost in text-video+audio retrieval performance.

2 Related Work

2.1 Large-Scale Video-Language Datasets

Manual annotation of video captioning datasets is extremely time-consuming since it involves video trimming and localization of caption boundaries. Currently, manually annotated datasets, e.g., MSR-VTT [58], YouCook2 [67], HIREST [64], and HT-Step [2], are limited in size. Therefore, different methods of mining videos with weak supervision from the Internet were considered. Datasets such as YouTube-8M [1] and IG-Kinetics-65M [15] provided multiple class labels based on query clicks, metadata [1] or hashtags [15]. However, short class labels are suboptimal supervision compared to textual descriptions [12]. Therefore, Bain et al. [4] considered scrapping videos with associated alt-text from the web, obtaining the WebVid2M and WebVid10M datasets [4] with 2M and 10M video-text pairs, respectively. Stroud et al. [48] proposed to use meta information, such as titles, descriptions, and tags from YouTube, as a textual

annotation and created the WTS-70M dataset. Nagrani et al. [34] proposed to transfer image captions from an image-text dataset to videos by searching videos with similar frames to an image and collected the VideoCC3M dataset. Yang et al. [61] created the VidChapters-7M dataset by scraping user-annotated chapters on YouTube. However, most videos in WebVid do not have audio, which is an essential part of video analysis, and captions in VideoCC3M are derived from images and, therefore, tend to describe more static scenes rather than actions. At the same time, the title, tags, and chapters of WTS-70M and VidChapters-7M provide only high-level video descriptions.

As an alternative to this, Miech et al. [33] proposed the HowTo100M dataset, where instructional videos are naturally accompanied by dense textual supervision in the form of subtitles obtained from ASR (Automatic Speech Recognition) systems. The HowTo100M dataset with 137M clips sourced from 1.2M YouTube videos was proven to be effective for pre-training video-audio-language representations [7, 44, 47]. The followed-up YT-Temporal-180M [65] and HD-VILA-100M [59] datasets follow the same idea but contain more videos with higher diversity and higher video resolution. While ASR supervision provides a scalable way to create large datasets with dense annotation, the quality of subtitles is not on par with human-annotated captions. In this work, we propose a method to create high-quality captions for videos at scale by leveraging LLM and subtitles.

Recent works, InternVid [56], VAST [9], and Video-ChatGPT [31], collect large-scale video-text datasets through per-frame image captioning and text summarization with LLMs. In contrast, our work stands apart in its objective. Rather than distilling existing knowledge from image models, pre-trained on human-annotated image-text data, into videos, we aim to gather a dataset enriched with new video knowledge. This is achieved by converting freely available ASR subtitles into captions. Concurrent work, HowToStep [23], summarizes ASR subtitles into descriptive steps using LLM and then performs additional temporal realignment. Our main difference is that we focus on general video captions rather than procedural steps.

2.2 Learning with Noisy ASR Subtitles of Instructional Videos

The problem of misalignment and noisiness of ASR supervision in instructional videos, e.g., in the HowTo100M dataset, were addressed in multiple works. MIL-NCE loss [32] and soft max-margin ranking loss [3] were proposed to adapt contrastive loss to misalignment in text-video pairs. Zellers et al. [65] proposed to use LLM to add punctuation and capitalization to ASR subtitles and remove mistranscription errors. Han et al. [16] proposed to train temporal alignment networks to filter out subtitles that are not alignable to the video and determine alignment for the others. However, to the best of our knowledge, [27] is the only work that goes beyond just removing mistranscription errors and ASR realignment, where Lin et al. proposed to match subtitles to step descriptions from WikiHow dataset [19] (distant supervision). In our work, we propose to use LLM to create video captions given ASR subtitles, which allows us to create detailed descriptions that are specific for every video and have proper sentence structure.

2.3 LLMs in Vision-Language Tasks

In recent years, there has been a remarkable success of LLMs in many language-related tasks [13,41,42]. Latest large language models [10,35,53,54] have demonstrated excellent zero-shot capabilities on common-sense inference [5]. This success has prompted research into integrating common-sense knowledge into vision-language tasks. In this regard, some methods [29,49–51] initialize the language part of vision-language models from pre-trained LLM. Another line of work [11, 21, 28, 66] uses LLM as a decoder to enable vision-to-language generation. Works [24, 66] adapted visually conditioned LLM for visual captioning and created captioning pseudo-labels for large-scale video data. However, these methods require human-annotated datasets to train a captioning model, while our method does not require any label data and aims to transform free available annotation (ASR subtitles) into textual descriptions.

3 Method

3.1 Problem Statement

Given a dataset of N untrimmed long-term instructional videos V_n with corresponding noisy ASR subtitles S_n, our goal is to create "human-written-like" video captions C_n (with $1 \leq n \leq N$). Note that our task does not assume access to any paired training data $((V_n, S_n), C_n)$. The goal is to create the video captions C_n in a *zero-shot* setting given only videos and subtitles (V_n, S_n). More formally, for each given video V_n, we also have a set of subtitles of spoken text in the video, $S_n = \{s_{n,j}, t^s_{n,j}, t^e_{n,j}\}_{j \leq |S_n|}$ with their start t^s and end timestamps t^e recognized by ASR-systems. For each video V_n, our goal is to generate dense captions and their timestamps $C_n = \{c_{n,i}, \tau^s_{n,i}, \tau^e_{n,i}\}_{i \leq |C_n|}$, where each caption $c_{n,i}$ describes a segment of the video, that starts at $\tau^s_{n,i}$ and ends at $\tau^e_{n,i}$.

The generated captions aim to serve for video-language or video-language-{other modalities (*e.g.*, audio)} tasks, providing language supervision in the form of human-style captions rather than scrambled noisy ASR subtitles. That enables the potential of collecting large-scale datasets with long-term videos and their dense textual descriptions for free, without human supervision.

3.2 Video-Language Retrieval Model

Before we describe our method for generating the HowToCaption dataset, we will briefly recap the video-language retrieval models (V-L model), as it is one of the main use cases for this dataset. Moreover, we also use a V-L model to improve the temporal alignment in the dataset.

We base our video-language retrieval model (V-L model) on the pre-trained BLIP image-language dual-encoder model [22]. We maintain the architecture of the text encoder $f(c) \in \mathbb{R}^d$ but, following CLIP4CLIP [30], adapt the image encoder $g(I)$ to a video encoder by averaging image embeddings obtained from uniformly sampled frames of the video: $g(V_n) = \sum_{I \in V_n} g(I) \in \mathbb{R}^d$. Dual-encoder

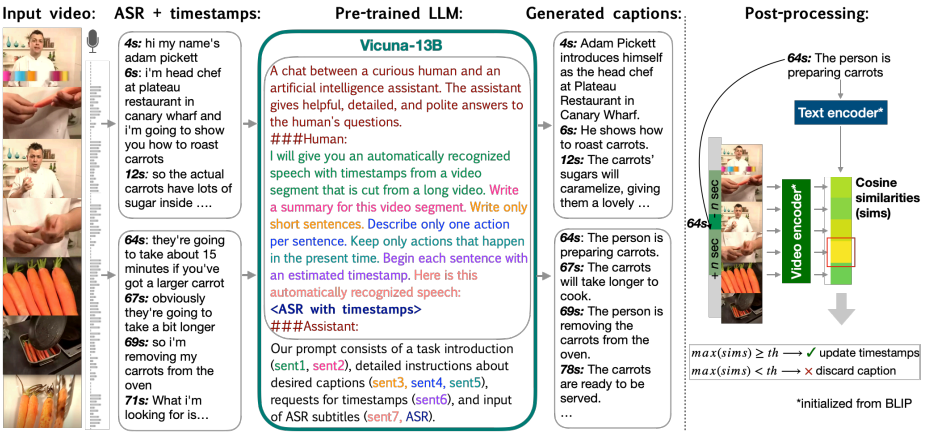


Fig. 2: Schematic visualization of the proposed HowToCaption method. Obtained from the automatic speech recognition system (ASR), subtitles are divided into blocks that contain longer contextual information. A large pre-trained language model is then used to generate plausible video captions based on ASR subtitles, along with timestamps for each caption. These generated captions and timestamps are further post-processed to enhance their alignment to the video and filter out captions with low similarity to the corresponding video by leveraging a pre-trained text-video model.

models typically learn a cross-modal embedding space [30, 39] via training with the symmetric InfoNCE loss [36]. The training is based on a similarity metric (often cosine distance) between embeddings $\rho_{n,i,m} = \text{sim}\left(f(c_{n,i}), g(V_m)\right)$ scaled by a temperature parameter ν, resulting in the following loss function:

$$L = -\frac{1}{2|B|} \sum_{(n,i)\in B} \left(\log \frac{\exp(\rho_{n,i,n}/\nu)}{\sum\limits_{(m,j)\in B} \exp(\rho_{n,i,m}/\nu)} + \log \frac{\exp(\rho_{n,i,n}/\nu)}{\sum\limits_{(m,j)\in B} \exp(\rho_{m,j,n}/\nu)} \right) \quad (1)$$

where B is a batch of training sample indices (n, i).

3.3 HowToCaption Method

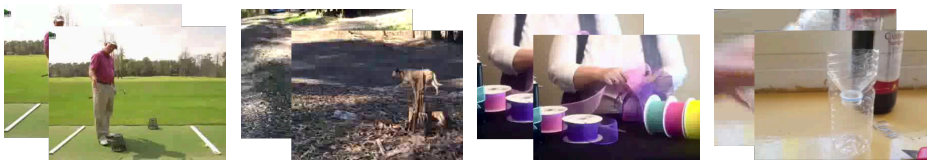
To generate captions for the instructional videos, we propose to leverage recent large language models that demonstrate great zero-shot performance in many different tasks formulated with natural language. Namely, we prompt the LLM to read the ASR subtitles of the video and create a plausible video description based on this. Since one subtitle only covers a small part of the video and lacks a global context, we propose to aggregate multiple subtitles together with their timestamp information. Then, we task the LLM to create detailed descriptions based on the entire input and estimate timestamps for each generated sentence.

The overview of our approach is shown in Fig. 2. For each video, we first slice a given sequence of subtitles into blocks that contain long context information

about the video. Then, the ASR subtitles of each block are summarised into a video caption using the LLM that we prompt with our task description. The LLM also predicts timestamps for each sentence in the video caption, which we further refine in our post-processing step based on similarities of a caption sentence to video clips in the neighboring area of predicted timestamps.

LLM Prompting. For our language prompt (shown in Fig. 2), we leverage the same "main" prompt for the LLM, as in the Vicuna-13B model [10]: "A chat between a curious human and..." that defines the requirement for LLM to give a helpful answer to our questions. Then, we describe our request, what data we need to process, and how it should be processed: "I will give you an automatically recognized speech...". We found structuring the prompt in the way that the task description given at the beginning of the prompt and the long ASR input S_n at the end is beneficial. Then, we give detailed instructions about how to process ASR subtitles. We found that instructions such as "Write only short sentences" or "Describe only one action per sentence" are beneficial, as they encourage the creation of concise captions that better match the video content. The instruction "Keep only actions that happen in the present time" is intended to filter out unrelated chats, advice, or comments from the captions; we observed that it also resulted in performance enhancements. Lastly, we request the model to predict a timestamp for each generated caption and, finally, input timestamps + ASR subtitles that need to be processed. The LLM response follows the start timestamp + caption format given in the prompt and, therefore, can be automatically parsed with a simple script into a set of captions and timestamps $C_n = \{c_{n,i}, \tau^s_{n,i}, \tau^e_{n,i}\}_{i \leq |C_n|}$, where we assign $\tau^e_{n,i} = \tau^s_{n,i} + \Delta_{sec}$, where Δ_{sec} is a constant video clip length parameter (number of seconds). Please see Sec. 4.3 and the supplement for a detailed evaluation of these choices. In the supplement, we also discuss an extension of our HowToCaption method to vision language models that additionally allows grounding captions on visual content. However, we found that this does not improve caption quality.

Post-processing: Alignment & Filtering. ASR subtitles often temporally misalign with video content [16, 33] (e.g., a speaker describes something only after it was shown in a video). Captions derived solely from ASR also face this issue. Therefore, inspired by the TAN method [16] that automatically predicts the alignability of subtitles and matching timestamps, we further improve our obtained captions with an alignment & filtering post-processing step (Fig. 2). To this end, we utilize the video-language encoder model (f, g). Given a generated caption $c_{n,i}$ and its start and end timestamps $(\tau^s_{n,i}, \tau^e_{n,i})$ that corresponds to a part of the video clip $V_n^{[\tau^s_{n,i}, \tau^e_{n,i}]}$, we use the V-L model to compute alignment similarity scores $\rho_{n,i}(\delta) = \text{sim}\left(f(c_{n,i}), g(V_n^{[\tau^s_{n,i}+\delta, \tau^e_{n,i}+\delta]})\right)$ between the caption and video clips with time offsets $\delta \in \mathbb{Z}, |\delta| \leq T$ around predicted timestamps. Then we *align* the caption to the video clip by finding the best offset around the timestamp $\delta^*_{n,i} = \underset{\delta \in \{-T,\ldots,T\}}{\arg\max}\ \rho_{n,i}(\delta)$ and *filter* out pairs if $\rho_{n,i}(\delta^*_{n,i}) < \kappa$, where κ is a similarity score threshold.



ASR: move them around to help direct the path
Caption: Matt Swanson gives a tip to use buckets to direct the path of the ball

ASR: so this is stage one of hiding the bone burying the bone ...
Caption: Dog wants to hang out near dirt or other dogs with bones to acquire more bones

ASR: so it's not going to really show
Caption: Making a bow with two colors

ASR: but this yeah and it just stays or it won't get off it's busy here
Caption: Make sure the bottle stays together

Fig. 3: Examples of video-captions pairs from our HowToCaption dataset. ASR subtitles with only noisy supervision for the video are converted from spoken- to written-language-style captions. Note that some details in the generated captions are taken from a longer context, see the supplement for a full example.

To further improve the alignment of captions, we perform multiple rounds of alignment & filtering. In practice, we found that the improvement after two rounds is marginal. For subsequent rounds, we fine-tune (c.f. Sec. 3.2) the V-L model on the aligned & filtered video-captions pairs $\{(v_i, c_i)\}$, resulting in new alignment scores $\rho'_{n,i}(\delta)$. Since fine-tuning V-L models often leads to forgetting, we employ two modifications in the fine-tuning and second alignment processes. First, during fine-tuning, we add regularization

$$L_{\text{align}} = \alpha \frac{1}{2|B|} \sum_{(n,i)\in B} (\text{sim}(f(c_{n,i}), f^*(c_{n,i})) + \text{sim}(g(V_n), g^*(V_n)) \qquad (2)$$

where f^* and g^* denote frozen text and video encoders, α is a regularization weight, and $(n, i) \in B$ represents the samples batch B. This regularization prevents the model from forgetting [17]. Then, during alignment & filtering, we use the average of the similarities of the fine-tuned and original model. We show an impact of these modifications in the supplement.

3.4 HowToCaption Dataset

We apply the proposed HowToCaption approach to 1.2M long-term instructional videos and ASR subtitles of the HowTo100M dataset and obtain the HowToCaption dataset. By prompting the Vicuna-13B model, we obtain $\sim$ 70M initial captions. The compute cost for prompting 1.2M videos is $\sim$12k GPU-hours on NVIDIA A40. After alignment & filtering (details in Sec. 4.2) we obtain 25M high-quality video-caption pairs. We show examples from our HowToCaption dataset in Fig. 3. We note that generated captions follow different text styles, *e.g.*, the first and the second examples contain a long description of an object and its actions, the third describes the process, and the last one is instruction. The average length of the captions is 9 words. We provide additional examples, statistics, failure case analyses, and user studies in the supplement.

Table 1: Ablation of LLM prompts. We step by step construct a prompt for an LLM that concisely and in detail describes the caption generation task. To emphasize our incremental adjustments, we label the sentences as xn (where n is an index). Each prompt consists of some sentences that were already used in previous prompt versions (e.g., <x1>, <x2>) and new sentences introduced in the current prompt (e.g., x4: Write only ...). With each prompt, we obtain 2M video-text pairs from 100k HowTo100M videos that we later use for T-V model training (lower-resource setup). Downstream zero-shot text-video retrieval performance is reported.

Prompt	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓
x1: Here is an automatically recognized speech from a video: <ASR with timestamps>. x2: Write a synopsis for this video. x3: Begin each sentence with an estimated timestamp.	37.5	22.5	71.0	**3**	80.5	**2**	37.3	30	56.6	14.4
<x1> <x2> x4: Write only short sentences. <x3>	39.3	20.5	71.4	**3**	81.0	**2**	36.5	32.5	57.1	14.5
<x1> <x2> <x4> x5: Describe only one action per sentence. <x3>	39.8	20	71.0	**3**	80.9	**2**	37.2	30.5	57.2	13.9
<x1> <x2> <x4> <x5> x6: Keep only actions that happen in the present time. <x3>	39.5	19.5	71.6	**3**	81.2	**2**	**37.9**	**29**	57.6	**13.4**
<x1> x2': Write *a summary* for this video. <x4> <x5> <x6> <x3>	40.4	**19**	71.4	**3**	81.4	**2**	37.1	30	57.6	13.5
x1': Here is an automatically recognized speech from *a video segment that is cut from a long video*: <ASR with timestamps> x2": Write a summary for *this video segment.* <x4> <x5> <x6> <x3>	40.0	20	**72.0**	**3**	81.1	**2**	37.8	**29**	57.7	13.5
I will give you an automatically recognized speech with timestamps from a video segment that is cut from a long video. <x2"> <x4> <x5> <x6> <x3> *Here is this automatically recognized speech: <ASR with timestamps>*	**40.6**	**19**	**72.0**	**3**	**81.6**	**2**	37.7	30	**58.0**	13.5

4 Experimental Results

To evaluate the proposed HowToCaption dataset for large-scale pre-training of vision-language models, we train a T-V model as described in Sec. 3.2 on the HowToCaption dataset and assess its zero-shot video-text retrieval performance on four widely recognized and diverse video-text benchmarks: YouCook2 [67], MSR-VTT [58], MSVD [8], and LSMDC [43]. While the YouCook2 dataset consists of instructional cooking videos and might be considered as an in-domain benchmark for the HowToCaption dataset, the other datasets encompass a broader range of topics and video types, including non-instructional YouTube videos and movies. To evaluate the properties of HowToCaption dataset in comparison with other large-scale pre-training datasets, we also train our T-V model on the HowTo100M [33], HowTo100M with step labels [27], HTM-AA [16], VideoCC3M [34], and WebVid2M [4] datasets and compare zero-shot text-video retrieval performance.

Table 2: Effect of a longer context. For the "no context" option, we predict captions from individual ASR subtitles. With our "long context" option, we input multiple ASR subtitles with timestamps and the model generated captions based on longer context. This ablation is done in lower-resource setup.

Method	YouCook2				MSR-VTT				MSVD				LSMDC				Average			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
No context	11.1	27.9	38.4	21	37.7	**62.4**	**72.6**	**3**	43.3	71.7	80.2	**2**	16.5	30.4	**38.4**	**30**	27.1	48.1	57.4	14
Long context	**12.1**	**30.0**	**40.6**	**19**	**37.9**	61.6	72	**3**	**43.9**	**72.7**	**81.6**	**2**	**16.8**	**31.4**	37.7	**30**	**27.7**	**48.9**	**58.0**	**13.5**

Table 3: Effect of alignment & filtering. With each post-processing variant, we obtain 25M video-text pairs that we later use for T-V model training. Downstream zero-shot text-video retrieval performance is reported.

Caption Post-processing	YouCook2		MSR-VTT		MSVD		LSMDC		Average	
	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓	R10↑	MR↓
Lower bound: original ASR as supervision	39.3	20	61.7	5	77.1	2	31.5	56	52.4	20.8
No post-processing	40.2	18	65.9	4	79.8	**2**	34.4	40	55.1	16.0
Filtering (using BLIP)	42.5	16	71.2	**3**	81.7	**2**	37.4	30	58.2	12.8
Alignment & filtering (using BLIP)	42.4	17	71.7	**3**	**82.2**	**2**	38.5	29.5	58.7	12.9
Alignment & filtering (with ours)	**44.1**	**15**	**73.3**	**3**	82.1	**2**	**38.6**	**29**	**59.5**	**12.3**

4.1 Datasets and Metrics

Pre-training Datasets. HowTo100M is a dataset of 1.2M instructional videos with ASR subtitles. We consider three versions of annotations of this dataset: *Sentencified HowTo100M* [16], with pre-processed ASR subtitles by structuring them into full sentences; *HowTo100M with Distant Supervision* [27], where ASR subtitles were linked to WikiHow [19] step descriptions via distant supervision; and *HTM-AA* [16], an auto-aligned (AA) version of HowTo100M, where subtitle timestamps were adjusted to improve alignment to videos and non-alignable subtitles were discarded. **WebVid2M** [4] is a large open-domain dataset of 2.5M of short videos scrapped from the internet with their alt-text. **VideoCC3M** [34] is a dataset of 10M video-text pairs collected by transferring captions from image-text CC3M dataset [6] to videos with similar visual content.

Datasets for Downstream Tasks. YouCook2 [67] is a dataset of instructional cooking videos, where each video clip is annotated with a recipe step. We used 3.5k test set for evaluation. **MSR-VTT** [58] contains 10k YouTube videos on various topics and human descriptions. Following prior work [4, 34, 52], we use the 1k test set for evaluation. **MSVD** [8] is a dataset of video snippets with their textual summary. The evaluation set consists of 670 videos with 40 captions corresponding to each video. We follow standard practice [4, 30] and count each caption-video pair towards the metrics. **LSMDC** [43] is a collection of video clips from movies with human-written descriptions. The test set consists of 1k video-caption pairs.

Metrics. To evaluate zero-shot text-video retrieval, we use standard Recall@K metrics where $K \in 1, 5, 10$ (R1, R5, R10) and Median Rank (MR).

4.2 Implementation Details

As an LLM, we utilize Vicuna-13B-v0 [10]. In the supplement, we additionally experiment with the MiniGPT-4 model [68] to generate captions from subtitles grounded on visual content. To create the HowToCaption dataset, we leverage subtitles with timestamps released by [16] (Sentencified HowTo100M), where officially released subtitles [33] for HowTo100M videos were post-processed by structuring them into full sentences. For our T-V model (described in Sec. 3.2), we use ViT-B/16 visual encoder and $\text{BERT}_{\text{base}}$ textual encoder that are initialized with $\text{BLIP}_{\text{CapFilt-L}}$ pre-trained weights. We uniformly sample 4 frames from a video clip during training and 12 frames during evaluation. For HowToCaption method, we use $T = 10$ seconds offset for alignment and adaptive threshold κ to leave 25M most similar pairs after filtering. Following [7] that found that 8-sec clips are optimal for training on HowTo100M, we set $\Delta_{\text{sec}} = 8$. More implementation details are in the supplement.

4.3 Ablation Studies

Prompt Engineering. First, we assess the quality of the obtained dataset with respect to different LLM prompts. Since prompting LLM with subtitles from 1.2M videos is resource-intensive, we perform the prompt engineering ablations in a lower-resource setup, where we use a 100k subset of HowTo100M ($\sim 10\%$ of all videos) to create dense captions with the LLM and use the threshold κ to obtain the 2M most confident video-caption pairs. Here, we train the T-V model for 150k iterations and then evaluate zero-shot on downstream tasks. In Tab. 1, we begin with a basic prompt for the LLM, gradually refining it to generate captions more suitable for vision-language tasks. It is essential to recognize that the impact of various prompts on performance can vary across datasets, as certain prompts may yield captions better aligned with specific downstream tasks. Notably, incorporating key phrases such as "Write only short sentences" or "Describe only one action per sentence" leads to performance improvements on 3 out of 4 datasets. Additionally, the use of the phrase "Keep only actions that happen in the present time" also results in performance enhancements. Furthermore, structuring the task description at the beginning and presenting the data to be processed at the end (the final modification) also boosts performance. We provide more ablations on prompt engineering in the supplement.

We also examine the impact of leveraging a longer context for caption prediction. In Tab. 2, we compare caption generation with "no context", where captions are predicted from individual ASR subtitles. With our "long context" option, we input multiple ASR subtitles with their timestamps, and the model predicts both captions and timestamps based on longer context. We found that using a longer context is beneficial, resulting in an average improvement of 0.6 p.p. in R10, and particularly advantageous for YouCook2 and MSVD.

Alignment & Filtering. Further, we assess the impact of the proposed alignment & filtering procedure on the quality of captions of the acquired dataset in Tab. 3. We examine the performance of the T-V model when trained on

Table 4: Zero-shot text-to-video retrieval performance of models trained on different video-text datasets. For each dataset, we train our T-V model and report downstream performance.

Video-Text Training Data	YouCook2				MSR-VTT				MSVD				LSMDC			
	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
- (zero-shot, with BLIP initialization)	6.1	16.2	23.6	69	34.3	59.8	70.6	3	38.5	65.0	74.0	2	14.7	29.5	36.5	31
HowTo100M with ASRs [16]	12.2	29.1	39.3	20	30.8	52.6	61.7	5	39.2	68.3	77.1	**2**	12.9	24.7	31.5	56
HowTo100M with distant supervision [27]	8.3	21.5	30.3	34	28.6	54.0	66.3	5	38.5	68.6	79.4	**2**	12.1	24.7	32.4	42.5
HTM-AA (auto-aligned ASRs) [16]	**13.4**	32.2	43.5	**15**	29.8	54.1	64.3	4	38.7	68.6	78.7	**2**	11.9	23.9	30.5	46
HowToCaption (ours)	**13.4**	**33.1**	**44.1**	**15**	37.6	**62.0**	**73.3**	**3**	**44.5**	73.3	**82.1**	**2**	17.3	**31.7**	38.6	29
VideoCC3M [34]	5.3	15.1	21.7	84	33.9	57.9	67.1	4	39.6	66.7	76.8	**2**	14.8	29.4	35.8	33
WebVid2M [4]	7.3	20.7	29.0	46	**38.5**	61.7	71.9	**3**	44.5	**73.4**	82.1	**2**	**17.8**	31.2	**39.8**	**25**

Table 5: Comparison in zero-shot text-to-video retrieval with *dual-encoder* baseline methods. [§]For BLIP, the performance of dual-encoder architecture is reported. [‡]6 datasets = CC3M [6]+CC12M [6]+COCO [26]+VG [20] +SBU [37] +LAION [45]. "V." = Vision, "I-T" = Image-Text, "V-T" = Video-Text.

Method	V. Encoder	I-T Data	V-T Data	YouCook2				MSR-VTT				MSVD				LSMDC			
				R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR	R1	R5	R10	MR
Nagrani et al. [34]	ViT-B	-	VideoCC3M	-	-	-	-	18.9	37.5	47.1	-	-	-	-	-	-	-	-	
Frozen-in-Time [4]	ViT-B/16	CC+COCO	WebVid2M	-	-	-	-	24.7	46.9	57.2	7								
CLIP-straight [38]	ViT-B/32	WIT	-	-	-	-	-	31.2	53.7	64.2	4	37.0	64.1	73.8	2	11.3	22.7	29.2	56.5
CLIP4CLIP [30]	ViT-B/32	WIT	HowTo100M	-	-	-	-	32.0	57.0	66.9	4	38.5	66.9	76.8	2	15.1	28.5	36.4	**28**
VideoCoCa [60]	ViT-B/18	JFT-3B	VideoCC3M	**16.5**	-	-	-	31.2	-	-	-	-	-	-	-	-	-	-	-
BLIP[§] [22]	ViT-B/16	6 datasets[‡]	-	6.1	16.2	23.6	69	34.3	59.8	70.6	**3**	38.5	65.0	74.0	2	14.7	29.5	36.5	30.5
BLIP[§]+HowTo100M	ViT-B/16	6 datasets[‡]	HowTo100M	12.2	29.1	39.3	20	30.8	52.6	61.7	5	39.2	68.3	77.1	2	12.9	24.7	31.5	56
Ours	ViT-B/16	6 datasets[‡]	HowToCaption	13.4	**33.1**	**44.1**	**15**	**37.6**	**62**	**73.3**	**3**	**44.5**	**73.3**	**82.1**	**2**	**17.3**	**31.7**	**38.6**	29

differently post-processed versions of the dataset. Remarkably, we discover that the obtained video-caption pairs, even without any post-processing, significantly outperform the original ASR-based supervision. Subsequently, by employing the alignment and filtering procedure to leave only 25M pairs based on video-caption similarities derived from BLIP pre-trained weights, we achieve a notable performance enhancement of 3.6 p.p. in R10. Furthermore, alignment & filtering with our proposed fine-tuning without forgetting yields an additional 0.8 p.p. boost in R10 performance. More ablations can be found in the supplement.

4.4 Comparison with State-of-the-art

Comparision with Other Web Datasets. In Tab. 4, we assess the pre-training effectiveness of our proposed HowToCaption dataset compared to other web video-language datasets. Specifically, we evaluate different textual annotations of HowTo100M videos: sentencified ASR subtitles [16], task steps from distant supervision [27], and auto-aligned ASR subtitles [16]. Additionally, we conduct evaluations on WebVid2M [4] and VideoCC3M [34] datasets. Our findings indicate that the model pre-trained on our HowToCaption dataset significantly outperforms models pre-trained on other versions of HowTo100M annotations, with an average improvement of 5.2 p.p. in R10. This improvement is most pronounced for the MSR-VTT, MSVD, and LSMDC datasets, which feature full-sentence captions. Interestingly, for the YouCook2 dataset with cap-

Table 6: Zero-shot text-to-video+audio retrieval. *Text-video only models.

Method	Vision Encoder	YouCook2				MSR-VTT			
		R1↑	R5↑	R10↑	MR↓	R1↑	R5↑	R10↑	MR↓
MIL-NCE* [32]	S3D	15.1	38.0	51.2	10	9.9	24.0	32.4	29.5
TAN* [16]	S3D	20.1	45.5	59.5	7.0	-	-	-	-
MMT [14]	Transformer	-	-	-	-	-	14.4	-	66
AVLNet [44]	ResNet-152+ResNeXt101	19.9	36.1	44.3	16	8.3	19.2	27.4	47
MCN [7]	ResNet-152+ResNeXt101	18.1	35.5	45.2	-	10.5	25.2	33.8	-
EAO [47]	S3D	24.6	48.3	60.4	6	9.3	22.9	31.2	35
Ours	S3D	**25.5**	**51.1**	**63.6**	**5**	**13.2**	**30.3**	**41.5**	**17**

Table 7: Video captioning results. We report BLEU@4 (B@4), METEOR (M), ROUGE-L (R), and CIDEr (C). We gray out methods that use a stronger vision backbone and significantly more pre-training data for fair comparison. ⋆4 datasets = CC3M+COCO+VG+SBU, †5 datasets = ...+CC12M, ‡6 datasets = ...+LAION [45]. §Our fine-tuning. "V." = Vision, "I-T" = Image-Text, "V-T" = Video-Text.

Method	V. Encoder	I-T Data	V-T Data	YouCook2				MSR-VTT				MSVD			
				B@4	M	R	C	B@4	M	R	C	B@4	M	R	C
SwinBERT [25]	VidSwin-B	-	-	**9.0**	15.6	37.3	109	41.9	29.9	62.1	53.8	58.2	41.3	77.5	120.6
CLIP4Caption [52]	ViT-B	-	-	-	-	-	-	46.1	30.7	63.7	57.7	-	-	-	-
GIT-B [55]	ViT-B	4 datasets⋆	-	5.8	12.2	31.5	80.3	46.6	29.6	63.2	57.8	69.3	44.5	81.4	142.6
MV-GPT [46]	ViViT-B	-	HowTo100M	-	-	-	-	48.9	**38.6**	64	60.0	-	-	-	-
LAVENDER	VidSwin-B	5 datasets†	WebVid2M+12M	-	-	-	-	-	-	-	60.1	-	-	-	150.7
HiTeA [63]	MViT-B	5 datasets†	WebVid2M	-	-	-	-	-	-	-	65.1	-	-	-	146.9
mPlug-2 [57]	ViT-B	5 datasets†	WebVid2M	-	-	-	-	**52.2**	32.1	**66.9**	**72.4**	69.3	45.1	81.9	148.2
BLIP [22]	ViT-B	6 datasets‡	-	7.9	15.0	36.0	104.8	49.2	32.2	66.1	65.5	68.0	45.3	82.8	148.6
BLIP + HowTo100M§	ViT-B	6 datasets‡	HowTo100M	8.6	**15.9**	37.1	112.9	49.4	32.0	66.2	65.9	69.3	46.2	83.0	151.4
Ours	ViT-B	6 datasets‡	HowToCaption	8.8	**15.9**	**37.3**	**116.4**	49.8	32.2	66.3	65.3	**70.4**	**46.4**	**83.2**	**154.2**
Vid2Seq [62]	ViT-L	-	YT-Temporal-1B	-	-	-	-	-	30.8	-	64.6	-	45.3	-	146.2
VideoCoCa [60]	ViT-G	JFT-3B	VideoCC3M	-	-	-	-	53.8	-	68.0	73.2	-	-	-	-
GIT-2 [55]	DaViT-4.8B	12.9B pairs	-	9.4	15.6	37.5	131.2	54.8	33.1	68.2	75.9	82.2	52.3	88.7	185.4

tions in the form of step descriptions like "cut tomato", HTM-AA already exhibits a high baseline performance, but our HowToCaption dataset still provides a performance boost. We also observe that the VideoCC3M dataset does not improve the initial BLIP performance on any datasets except for the MSVD. We attribute it to the fact that the VideoCC3M dataset adopts captions from the CC3M dataset [6] and transfers them to videos, potentially not introducing significantly new knowledge for the BLIP-initialised model since BLIP was pre-trained on multiple datasets, including CC3M. On the other hand, WebVid2M demonstrated performance improvements across all datasets, but our HowToCaption dataset notably outperforms WebVid2M on YouCook2 and MSR-VTT, only underperforming on LSMDC.

Comparison with State-of-the-art in Zero-shot Text-Video Retrieval. In Tab. 5, we also conduct a comparison with zero-shot *dual-encoder* retrieval baselines. It is important to acknowledge that comparing state-of-the-art methods can be challenging due to variations in backbone capacity, training objectives, and other factors. Therefore, we focus on a comparison with dual-encoder models in the zero-shot settings. Additional comparisons with methods that use *re-ranking* can be found in the supplement. Nevertheless, it is worth highlighting

that our approach consistently outperforms the baseline methods in zero-shot text-video retrieval across all datasets.

Zero-shot Text-to-Video+Audio Retrieval. It is known that instructional video datasets, *e.g.*, HowTo100M, suffer from a high correlation of audio modality to a textual description, therefore hindering building a text-video+audio retrieval system where the video is extended with audio [34, 47]. The usage of ASR narrations as supervisory textual description leads retrieval models to primarily perform speech recognition on the audio, hindering true language-audio connections [47]. Therefore, training text-video+audio systems on such datasets usually requires additional regularization, such as shifting audio timestamps or assigning lower weights to the audio loss [47]. Our HowToCaption dataset resolves this issue by providing richer textual descriptions, allowing us to train a text-video+audio retrieval system without regularization. To evaluate this, we train a multimodal Everything-At-Once (EAO) [47] model that learns to fuse any combinations of text, video, and audio modalities on our proposed HowToCaption dataset without any additional regularization and evaluate zero-shot text-video+audio retrieval performance. Tab. 6 shows the proposed model significantly outperforms all baselines and the directly comparable EAO model.

Video Captioning. We further validate our dataset's potential for video-language pre-training through video captioning. We fine-tune the BLIP [22] model with all architecture blocks, including the image-grounded text decoder, on the HowToCaption dataset with all three BLIP losses: image-text contrastive loss, image-text matching loss, and language modeling loss. We adapt the image encoder to the video encoder similarly to our retrieval model (Sec. 3.2) and use similar hyperparameters (see the supplement). Subsequently, we fine-tune the model with language modeling loss on the training set of either YouCook2, MSR-VTT, or MSVD datasets and evaluate video captioning performance. Tab. 7 show that our model fine-tuned on the HowToCaption dataset significantly outperforms directly comparable BLIP model and other comparable baselines on YouCook2 and MSVD and keeps competitive performance on MSR-VTT.

5 Conclusion

In this work, we propose a novel approach, HowToCaption, that transforms freely available ASR subtitles of instructional videos into high-quality captions, enabling the collection of large-scale, high-quality video-language datasets without manual annotation efforts. With this approach, we curate a new large-scale HowToCaption dataset featuring human-style video captions derived from ASR subtitles of the HowTo100M dataset. We demonstrate that our HowToCaption dataset serves as an excellent source for training video-language representation and video captioning models across four text-video retrieval and three video captioning benchmarks. Furthermore, given the separation of textual descriptions from the audio modality in our dataset, it demonstrates efficacy as a valuable resource for text-to-video-audio tasks. This work demonstrates the potential of LLMs for creating annotation-free, large-scale video-language datasets.

Acknowledgements

Nina Shvetsova is supported in part by the German Federal Ministry of Education and Research (BMBF) project STCL - 01IS22067.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675 (2016)
2. Afouras, T., Mavroudi, E., Nagarajan, T., Wang, H., Torresani, L.: Ht-step: Aligning instructional articles with how-to videos. NeurIPS **36** (2024)
3. Amrani, E., Ben-Ari, R., Rotman, D., Bronstein, A.: Noise estimation using density estimation for self-supervised multimodal learning. AAAI (2021)
4. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV (2021)
5. Chang, T.A., Bergen, B.K.: Language model behavior: A comprehensive survey. arXiv preprint arXiv:2303.11504 (2023)
6. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR (2021)
7. Chen, B., Rouditchenko, A., Duarte, K., Kuehne, H., Thomas, S., Boggust, A., Panda, R., Kingsbury, B., Feris, R., Harwath, D., et al.: Multimodal clustering networks for self-supervised learning from unlabeled videos. In: ICCV (2021)
8. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL (2011)
9. Chen, S., Li, H., Wang, Q., Zhao, Z., Sun, M., Zhu, X., Liu, J.: Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. NeurIPS **36** (2023)
10. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. Large Model Systems Organization (2023)
11. Cho, J., Lei, J., Tan, H., Bansal, M.: Unifying vision-and-language tasks via text generation. In: ICML (2021)
12. Desai, K., Johnson, J.: Virtex: Learning visual representations from textual annotations. In: CVPR (2021)
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
14. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: ECCV (2020)
15. Ghadiyaram, D., Tran, D., Mahajan, D.: Large-scale weakly-supervised pre-training for video action recognition. In: CVPR (2019)
16. Han, T., Xie, W., Zisserman, A.: Temporal alignment networks for long-term video. In: CVPR (2022)
17. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR (2019)
18. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)

19. Koupaee, M., Wang, W.Y.: Wikihow: A large scale text summarization dataset. arXiv preprint arXiv:1810.09305 (2018)
20. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV (2017)
21. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
22. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
23. Li, Z., Chen, Q., Han, T., Zhang, Y., Wang, Y., Xie, W.: A strong baseline for temporal video-text alignment. arXiv preprint arXiv:2312.14055 (2023)
24. Lialin, V., Rawls, S., Chan, D., Ghosh, S., Rumshisky, A., Hamza, W.: Scalable and accurate self-supervised multimodal representation learning without aligned video and text data. In: WACV (2023)
25. Lin, K., Li, L., Lin, C.C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y., Wang, L.: Swinbert: End-to-end transformers with sparse attention for video captioning. In: CVPR (2022)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
27. Lin, X., Petroni, F., Bertasius, G., Rohrbach, M., Chang, S.F., Torresani, L.: Learning to recognize procedural activities with distant supervision. In: CVPR (2022)
28. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. NeurIPS **36** (2023)
29. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: NeurIPS (2019)
30. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. Neurocomputing (2022)
31. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv preprint arXiv:2306.05424 (2023)
32. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-end learning of visual representations from uncurated instructional videos. In: CVPR (2020)
33. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: ICCV (2019)
34. Nagrani, A., Seo, P.H., Seybold, B., Hauth, A., Manen, S., Sun, C., Schmid, C.: Learning audio-video modalities from image captions. In: ECCV (2022)
35. Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J.M., Tworek, J., Yuan, Q., Tezak, N., Kim, J.W., Hallacy, C., et al.: Text and code embeddings by contrastive pre-training. arXiv preprint arXiv:2201.10005 (2022)
36. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
37. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. In: NeurIPS (2011)
38. Portillo-Quintero, J.A., Ortiz-Bayliss, J.C., Terashima-Marín, H.: A straightforward framework for video retrieval using clip. In: Pattern Recognition: 13th Mexican Conference (2021)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)

40. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: ICML (2023)
41. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog (2019)
42. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR (2020)
43. Rohrbach, A., Rohrbach, M., Schiele, B.: The long-short story of movie description. In: GCPR. Springer (2015)
44. Rouditchenko, A., Boggust, A., Harwath, D., Chen, B., Joshi, D., Thomas, S., Audhkhasi, K., Kuehne, H., Panda, R., Feris, R., Kingsbury, B., Picheny, M., Torralba, A., Glass, J.: Avlnet: Learning audio-visual language representations from instructional videos. In: Interspeech (2021)
45. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021)
46. Seo, P.H., Nagrani, A., Arnab, A., Schmid, C.: End-to-end generative pretraining for multimodal video captioning. In: CVPR (2022)
47. Shvetsova, N., Chen, B., Rouditchenko, A., Thomas, S., Kingsbury, B., Feris, R.S., Harwath, D., Glass, J., Kuehne, H.: Everything at once-multi-modal fusion transformer for video retrieval. In: CVPR (2022)
48. Stroud, J.C., Lu, Z., Sun, C., Deng, J., Sukthankar, R., Schmid, C., Ross, D.A.: Learning video representations from textual web supervision. arXiv preprint arXiv:2007.14937 (2020)
49. Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., Dai, J.: Vl-bert: Pre-training of generic visual-linguistic representations. In: ICLR (2020)
50. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV (2019)
51. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: EMNLP (2019)
52. Tang, M., Wang, Z., Liu, Z., Rao, F., Li, D., Li, X.: Clip4caption: Clip for video caption. In: ACMMM (2021)
53. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models (2023)
54. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
55. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., Wang, L.: Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 (2022)
56. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. arXiv preprint arXiv:2307.06942 (2023)
57. Xu, H., Ye, Q., Yan, M., Shi, Y., Ye, J., Xu, Y., Li, C., Bi, B., Qian, Q., Wang, W., et al.: mplug-2: A modularized multi-modal foundation model across text, image and video. arXiv preprint arXiv:2302.00402 (2023)
58. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR (2016)

59. Xue, H., Hang, T., Zeng, Y., Sun, Y., Liu, B., Yang, H., Fu, J., Guo, B.: Advancing high-resolution video-language representation with large-scale video transcriptions. In: CVPR (2022)
60. Yan, S., Zhu, T., Wang, Z., Cao, Y., Zhang, M., Ghosh, S., Wu, Y., Yu, J.: Video-text modeling with zero-shot transfer from contrastive captioners. arXiv preprint arXiv:2212.04979 (2022)
61. Yang, A., Nagrani, A., Laptev, I., Sivic, J., Schmid, C.: Vidchapters-7m: Video chapters at scale. NeurIPS **36** (2024)
62. Yang, A., Nagrani, A., Seo, P.H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J., Schmid, C.: Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In: CVPR (2023)
63. Ye, Q., Xu, G., Yan, M., Xu, H., Qian, Q., Zhang, J., Huang, F.: Hitea: Hierarchical temporal-aware video-language pre-training. In: ICCV. pp. 15405–15416 (2023)
64. Zala, A., Cho, J., Kottur, S., Chen, X., Oguz, B., Mehdad, Y., Bansal, M.: Hierarchical video-moment retrieval and step-captioning. In: CVPR (2023)
65. Zellers, R., Lu, X., Hessel, J., Yu, Y., Park, J.S., Cao, J., Farhadi, A., Choi, Y.: Merlot: Multimodal neural script knowledge models. NeurIPS (2021)
66. Zhao, Y., Misra, I., Krähenbühl, P., Girdhar, R.: Learning video representations from large language models. In: CVPR (2023)
67. Zhou, L., Xu, C., Corso, J.: Towards automatic learning of procedures from web instructional videos. In: AAAI (2018)
68. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)