HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips

Antoine Miech[1,2*] Dimitri Zhukov[1,2*] Jean-Baptiste Alayrac[2+]
Makarand Tapaswi[2] Ivan Laptev[1,2] Josef Sivic[1,2,3]
[1]École Normale Supérieure [2]Inria [3]CIIRC, CTU
https://www.di.ens.fr/willow/research/howto100m

Abstract

Learning text-video embeddings usually requires a dataset of video clips with manually provided captions. However, such datasets are expensive and time consuming to create and therefore difficult to obtain on a large scale. In this work, we propose instead to learn such embeddings from video data with readily available natural language annotations in the form of automatically transcribed narrations. The contributions of this work are three-fold. First, we introduce HowTo100M*: a large-scale dataset of 136 million video clips sourced from 1.22M narrated instructional web videos depicting humans performing and describing over 23k different visual tasks. Our data collection procedure is fast, scalable and does not require any additional manual annotation. Second, we demonstrate that a text-video embedding trained on this data leads to state-of-the-art results for text-to-video retrieval and action localization on instructional video datasets such as YouCook2 or CrossTask. Finally, we show that this embedding transfers well to other domains: fine-tuning on generic Youtube videos (MSR-VTT dataset) and movies (LSMDC dataset) outperforms models trained on these datasets alone. Our dataset, code and models are publicly available [1].*

1. Introduction

Communicating about the visual world using language is a key ability of humans as intelligent beings. A three year old child can manipulate objects, observe its own actions and describe them to others using language; while adults can learn new skills by reading books or watching videos. This interplay between video and language extends naturally to artificial agents that need to understand the visual world and communicate about it with people. Examples of tasks that still represent a significant challenge for current artificial systems include text-to-video retrieval [25, 32, 54, 55, 63], text-based action or event localization [15], video captioning [36, 61], and video question answering [51, 63]. Yet, progress on these problems is important for a host of applications from searching video archives to human-robot communication.

Figure 1: We learn a joint text-video embedding by watching millions of narrated video clips of people performing diverse visual tasks. The learned embedding transfers well to other instructional and non-instructional text-video datasets.

A common approach to model visual concepts described with language is to learn a mapping of text and video into a shared embedding space, where related text fragments and video clips are close to each other [15, 32, 37, 38, 59]. Learning a good representation often requires a large set of paired video clips and text captions. In fact, given the huge variability of video scenes and their textual descriptions, learning a generic embedding space may require millions of paired video clips and text captions. However, existing datasets (*e.g.* MSR-VTT [58], DiDeMo [15], EPIC-

*Equal contribution.
+Now at DeepMind.
[1]Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France.
[3]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

KITCHENS [7]), are on the scale of tens to hundreds of thousands of such pairs that have been annotated manually. Manual collection of such datasets is expensive and hard to scale. It is also subjective since video annotation can often be an ill-defined task with low annotator consistency [58].

In this work, we explore a different source of supervision to obtain paired video clips and text captions for learning joint representations of video and language. We observe that *narrated instructional videos* are available in large quantities (*e.g*. on YouTube) and provide a large amount of visual and language data. In particular, instructional videos [2, 30, 68] often contain narration with an explicit intention of explaining the visual content on screen. To leverage this rich source of data, we collect a new large-scale dataset containing 136 million video clips sourced from 1.22 million narrated instructional videos depicting humans performing more than 23,000 different tasks. Each clip is paired with a text annotation in the form of an automatically transcribed narration.

Contributions. The contributions of this work are threefold. First, we collect a new dataset of close-captioned video clips, *HowTo100M*, that is orders of magnitude larger than any other existing video-text datasets (Section 3). Second, we show that such data can be used to learn powerful video-language representations. Our model (Section 4), trained on HowTo100M, sets a new state-of-the-art for text-based action localization and text-to-video retrieval on existing datasets of instructional videos, YouCook2 [67] and CrossTask [68]. Finally, we explore the ability of models trained on our data to transfer to non-instructional videos. In particular, we demonstrate that models pretrained on HowTo100M can be successfully transferred by fine tuning on the MSR-VTT dataset (generic Youtube videos) and the LSMDC dataset (movies).

2. Related work

A significant number of computer vision applications rely on a joint understanding of visual and textual cues. These applications include automatic image and video captioning [20, 36, 60, 61], visual question answering [9, 29, 51, 63], visual content retrieval based on textual queries [32, 56, 63], temporal localization of events in videos using natural language [15, 26] or video summarization with natural language [38].

Vision, language and speech. A common approach to model vision and language is learning a joint embedding space where visual and textual cues are adjacent if and only if they are semantically similar [6, 8, 10, 11, 25, 32, 35, 37, 38, 59, 54, 55, 57]. Most of these works rely on medium scale well annotated datasets in which descriptive captions are collected for each video clip. This process is costly as it requires considerable human annotation effort making these datasets hard to scale (see Table 1). In this work, we train a joint video and language model *without a single manually annotated video description* by leveraging automatically transcribed narrated videos. Using the spoken text from narrated videos to supervise vision models has seen some recent interest [2, 5, 13, 30, 45, 62]. Harwath *et al*. [13] utilize the raw speech waveform to supervise the visual model, however, their method does not scale as annotators were paid to record audio descriptions for thousands of images. Chen *et al*. [5] use subtitles from documentaries to automatically obtain object labels, but their focus is on learning object detectors rather than text-video embeddings and their dataset contains only 9 documentary movies, compared to about 15 years of video content considered in this work.

Dataset	Clips	Captions	Videos	Duration	Source	Year
Charades [48]	10k	16k	10,000	82h	Home	2016
MSR-VTT [58]	10k	200k	7,180	40h	Youtube	2016
YouCook2 [67]	14k	14k	2,000	176h	Youtube	2018
EPIC-KITCHENS [7]	40k	40k	432	55h	Home	2018
DiDeMo [15]	27k	41k	10,464	87h	Flickr	2017
M-VAD [52]	49k	56k	92	84h	Movies	2015
MPII-MD [43]	69k	68k	94	41h	Movies	2015
ANet Captions [26]	100k	100k	20,000	849h	Youtube	2017
TGIF [27]	102k	126k	102,068	103h	Tumblr	2016
LSMDC [44]	128k	128k	200	150h	Movies	2017
How2 [45]	185k	185k	13,168	298h	Youtube	2018
HowTo100M	**136M**	**136M**	**1.221M**	**134,472h**	Youtube	2019

Table 1: Comparison of existing video description datasets. The size of our new HowTo100M dataset bypasses the size of largest available datasets by three orders of magnitude. M denotes million while k denotes thousand.

Learning from instructional videos. Instructional videos are rising in popularity in the context of learning steps of complex tasks [2, 16, 41, 42, 46, 68], visual-linguistic reference resolution [17, 18], action segmentation in long untrimmed videos [66] and joint learning of object states and actions [3]. Related to our work, [2, 30, 62] also consider automatically generated transcription of narrated instructional videos as a source of supervision. However as opposed to our work, these works typically extract from transcriptions only a small number of predefined labels.

Numerous datasets of web instructional videos were proposed over the past years [2, 30, 45, 47, 50, 67, 68]. Among the first to harvest instructional videos, Sener *et al*. [47] use WikiHow, an encyclopedia of *how to* articles, to collect 17 popular physical tasks, and obtain videos by querying these tasks on YouTube. In a similar vein, COIN [50] and CrossTask [68] datasets are collected by first searching for tasks on WikiHow and then videos for each task on YouTube. We use the same approach for collecting HowTo100M. The main distinction between our dataset and previous efforts is the unprecedented scale both in terms of variety (more than 23,000 tasks from 12 different domains) and size (136 million clips sourced from 1.2 million instruc-

Figure 2: Examples of clip-caption pairs retrieved with the help of our joint embedding. Pairs are selected based on the similarity between visual appearance and corresponding narration, while they are arranged based on linguistic similarity across pairs. Examples are taken from 4 distinct clusters, corresponding to *Knitting*, *Woodwork/Measuring*, *Cooking/Seasoning* and *Electric maintenance*.

tional videos).

Large scale data for model pretraining. The use of large scale and potentially noisy data from the web is an exciting prospect to pretrain language and vision models. In natural language processing, BERT [19], GPT [39], and GPT-2 [40] are examples of language models trained on large-scale data that achieve state-of-the-art for many tasks. In fact, training GPT-2 on WebText [40] a dataset of 40GB of text from Reddit achieves state-of-the-art even in *zero-shot* settings. In vision, [28, 49] explore the use of image metadata such as Instagram hashtags to pretrain image classifiers.

We are inspired by these works and focus our efforts on learning a strong embedding for joint understanding of video and language. We demonstrate that our video-language embedding learned from millions of YouTube videos not only outperforms previous work on tasks related to instructional videos without fine-tuning, but also generalizes well to non-instructional videos with some fine-tuning. We release our dataset, feature extraction pipeline, and model parameters as a resource that the video and language community can build on.

3. The HowTo100M dataset

We collect a new dataset of narrated videos with an emphasis on instructional videos where content creators teach complex tasks. This ensures that most narrations describe the observed visual content. HowTo100M features 1.22 million videos from YouTube, with activities from domains such as cooking, hand crafting, personal care, gardening, *etc*. Each video is associated with a narration available as subtitles that are either written manually or are the output of an Automatic Speech Recognition (ASR) system.

3.1. Data collection

Visual tasks. With an aim to obtain instructional videos that describe how to perform certain activities, we first start by acquiring a large list of activities using *WikiHow*[1] – an online resource that contains 120,000 articles on *How to ...* for a variety of domains ranging from cooking to human relationships structured in a hierarchy. We are primarily interested in "visual tasks" that involve some interaction with the physical world (*e.g. Making peanut butter*, *Pruning a tree*) as compared to others that are more abstract (*e.g. Ending a toxic relationship*, *Choosing a gift*). To obtain predominantly visual tasks, we limit them to one of 12 categories (listed in Table 2). We exclude categories such as *Relationships* and *Finance and Business*, that may be more abstract.

We further refine the set of tasks, by filtering them in a semi-automatic way. In particular, we restrict the primary verb to physical actions, such as *make*, *build* and *change*, and discard non-physical verbs, such as *be*, *accept* and *feel*. This procedure yields 23,611 visual tasks in total.

Instructional videos. We search for YouTube videos related to the task by forming a query with *how to* preceding the task name (*e.g. how to paint furniture*). We choose videos that have English subtitles - either uploaded manually, generated automatically by YouTube ASR, or generated automatically after translation from a different language by YouTube API.

We improve the quality and consistency of the dataset, by adopting the following criteria. We restrict to the top 200 search results, as the latter ones may not be related to the query task. Videos with less than 100 views are removed as they are often of poor quality or are amateurish. We also

[1] https://www.wikihow.com

Category	Tasks	Videos	Clips
Food and Entertaining	11504	497k	54.4M
Home and Garden	5068	270k	29.5M
Hobbies and Crafts	4273	251k	29.8M
Cars & Other Vehicles	810	68k	7.8M
Pets and Animals	552	31k	3.5M
Holidays and Traditions	411	27k	3.0M
Personal Care and Style	181	16k	1.6M
Sports and Fitness	205	16k	2.0M
Health	172	15k	1.7M
Education and Communications	239	15k	1.6M
Arts and Entertainment	138	10k	1.2M
Computers and Electronics	58	5k	0.6M
Total	23.6k	1.22M	136.6M

Table 2: Number of tasks, videos and clips within each category.

ignore videos that have less than 100 words as that may be insufficient text to learn a good video-language embedding. Finally, we remove videos longer than 2,000 seconds.

As some videos may appear in several tasks, we deduplicate videos based on YouTube IDs. However, note that the dataset may still contain duplicates if a video was uploaded several times or edited and re-uploaded. Nevertheless, this is not a concern at our scale.

3.2. Paired video clips and captions

Subtitles are often organized as a list of text chunks (lines), and need not form complete sentences. Each line is associated with a time interval in the video, typically the duration in which the line is uttered. We select each line of the subtitles as a caption, and pair it with the video clip from the time interval corresponding to the line. We show some examples from our clip-caption pairs in Figure 2.

Different from other datasets with clip-caption pairs (*e.g*. MSR-VTT), our captions are *not* manually annotated, but automatically obtained through the narration. Thus, they can be thought of as *weakly paired*. Typical examples of incoherence include the content producer asking viewers to subscribe to their channel, talking about something unrelated to the video, or describing something before or after it happens. Furthermore, our captions are often incomplete, lack punctuation, or are grammatically incorrect sentences, as they come from continuous narration and often ASR. We have manually inspected 400 randomly sampled clip-caption pairs and found that in 51 %, at least one object or action mention in the caption is visually seen in the video clip.

Statistics. The initial set of visual tasks are obtained by focusing on 12 WikiHow categories. Table 2 shows the number of collected WikiHow tasks and corresponding videos and clips per category. In Appendix A [33], we show the first two levels of the WikiHow hierarchy: the twelve categories and their subcategories along with the number of chosen tasks and corresponding videos in our dataset. We compare the sizes of existing clip-caption paired datasets in Table 1. HowTo100M is several orders of magnitude larger than existing datasets and contains an unprecedented duration (15 years) of video data. However, unlike previous datasets, HowTo100M does not have clean annotated captions. As the videos contain complex activities, they are relatively long with an average duration of 6.5 minutes. On average, a video produces 110 clip-caption pairs, with an average duration of 4 seconds per clip and 4 words (after excluding stop-words) per caption. For more details, we show in Appendix A [33] the distribution of nouns and verbs. Our data collection procedure assumes that searching with *How to* queries on YouTube would result in mostly instructional videos. We verify this by randomly selecting 100 videos and labeling their type. 71% of the videos are found to be instructional, 12% are vlogs, and another 7% are product reviews or advertisements. Note that vlogs, reviews and ads may also contain correspondences between visual content and narration. In particular, we noticed that objects shown on screen are often mentioned in narration. We do not discard such non-instructional videos, as they may still be useful for the learning the joint embedding.

4. Text-video joint embedding model

We now present our model to learn a joint text-video embedding from the automatically paired video clips and captions in our dataset. More formally, we are given a set of n video clips and associated captions $\{(V_i, C_i)\}_{i=1}^n$. We denote by $\mathbf{v} \in \mathbb{R}^{d_v}$ and $\mathbf{c} \in \mathbb{R}^{d_c}$ the d_v and d_c dimensional feature representation of a video clip V and caption C, respectively. Given this, our goal is to learn two mapping functions: $f : \mathbb{R}^{d_v} \rightarrow \mathbb{R}^d$ and $g : \mathbb{R}^{d_c} \rightarrow \mathbb{R}^d$ that respectively embed video and caption features into a common d-dimensional space, such that the cosine similarity

$$s(V, C) = \frac{\langle f(\mathbf{v}), g(\mathbf{c}) \rangle}{\|f(\mathbf{v})\|_2 \|g(\mathbf{c})\|_2} \quad (1)$$

is high when caption C describes the video clip V, and low otherwise.

In this work, we use the class of non-linear embedding functions used in [32], which are given by:

$$f(\mathbf{v}) = (W_1^v \mathbf{v} + b_1^v) \circ \sigma(W_2^v (W_1^v \mathbf{v} + b_1^v) + b_2^v) \quad (2)$$

$$\text{and } g(\mathbf{c}) = (W_1^c \mathbf{c} + b_1^c) \circ \sigma(W_2^c (W_1^c \mathbf{c} + b_1^c) + b_2^c), \quad (3)$$

where $W_1^v \in \mathbb{R}^{d \times d_v}$, $W_1^c \in \mathbb{R}^{d \times d_c}$, $W_2^v, W_2^c \in \mathbb{R}^{d \times d}$, $b_1^v, b_1^c, b_2^v, b_2^c \in \mathbb{R}^d$ are learnable parameters, σ is an element-wise sigmoid activation and $\circ$ is the element-wise multiplication (Hadamard product). In practice, $d_v = 4,096$, $d_c = 4,096$ and $d = 4,096$ resulting in a model composed of 67M parameters. Note that the first term on the right-hand side in Equations (2) and (3) is a linear fully-connected layer and the second term corresponds to a context gating function [31] with an output ranging between 0 and 1, which role is to modulate the output of the linear layer. As a result, this embedding function can model non-linear multiplicative interactions between the dimensions of the input feature vector which has proven effective in other text-video embedding applications [32].

Loss. We train our embedding model using the max-margin ranking loss [21, 32, 54, 55, 64]. At each iteration of our training algorithm, we sample a mini-batch $\mathcal{B} = \{i_1, ..., i_b\} \subset \{1, \dots, n\}$ of caption-clip training pairs $(V_i, C_i)_{i \in \mathcal{B}}$, and update the model parameters with a gradient step of the following loss:

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}(i)} \max(0, \delta + s_{i,j} - s_{i,i}) + \max(0, \delta + s_{j,i} - s_{i,i}),$$

where $s_{i,j} = s(V_i, C_j)$ is the similarity score (1) between video clip V_i and caption C_j, $\mathcal{N}(i)$ is a set of negative pairs for caption-clip i and δ is the margin. The first term in Equation (B) corresponds to the ranking loss when sampling a negative caption, while the second term corresponds to sampling a negative video clip. We fix $\delta = 0.1$ in practice. Our model parameters are updated using Adam [23] with a learning rate of 10^{-4}. Implementation details of the loss are provided in Appendix B [33].

Sampling strategy. Similar to [15], we apply an intra-video negative sampling strategy to define $\mathcal{N}(i)$. We show in Section 5.3 that this approach is *critical* for good performance. More precisely, half of our negative pairs $\{(V_i, C_j) : i \neq j\}$, are selected such that the video clip V_i and the caption C_j belong to the same original YouTube video (as (V_i, C_i)), while the other half are sampled from other YouTube videos. We apply intra-negative sampling to ensure that the learned embedding focuses on relevant aspects of the video clip (*e.g.* the hands of the person showing how to knead dough) rather than irrelevant background features (*e.g.* the kitchen). In Appendix C [33], we also provide an empirical analysis of the positive pair sampling strategy. We show that even though the training data is noisy, our attempts to automatically select correct positive pairs during training did not yield improvements so far. We think this could be attributed to the fact our model is shallow and is trained on a large amount of data.

Clip and caption representation. The clip feature $\mathbf{v}$ consists of temporally max-pooled pre-extracted CNN features. The caption feature $\mathbf{c}$ is the output of a shallow 1D-CNN on top of pre-computed word embeddings. More details are given in Section 5.1.

5. Experiments

In this section, we demonstrate that a strong joint representation for video and text can be learned from our unlabeled HowTo100M dataset. We provide experimental results for a variety of domains ranging from instructional videos in CrossTask [68], cooking videos in YouCook2 [67], generic YouTube videos in MSR-VTT [58] to movie video clips in LSMDC [44]. Specifically, we evaluate our learned embedding on the tasks of localizing steps in instructional videos of CrossTask [68] and text-based video retrieval on YouCook2 [67], MSR-VTT [58] and LSMDC [44] datasets.

Negative sampling	M (R@10)	L (R@10)	Y (R@10)	C (AVG Recall)
No intra-negative	**30.1**	12.3	18.1	25.7
With intra-negative	29.6	**14.0**	**24.8**	**33.6**

Table 3: Impact of intra-video negative pairs during training. M: MSR-VTT, L: LSMDC, Y: YouCook2, C: CrossTask.

Our *key* findings are the following: **(i)** For instructional video datasets, such as CrossTask [68] and YouCook2 [67], our off-the-shelf embedding trained on HowTo100M significantly outperforms state-of-the-art models trained on much smaller and manually-annotated datasets. **(ii)** On generic YouTube videos (MSR-VTT [58]), our HowTo100M embedding provides competitive retrieval performance compared to state-of-the-art methods trained on MSR-VTT. Moreover, we show that fine-tuning our pre-trained embedding model on just a fifth of annotated videos from MSR-VTT outperforms state-of-the-art. **(iii)** We show that fine-tuning our embedding on LSMDC enables generalization to movie videos and scripts despite the large domain gap. **(iv)** Finally, we demonstrate the importance of scale in HowTo100M to learn better joint video-text embeddings.

5.1. Implementation details

Video features. We extract frame-level and video-level features with pre-trained 2D and 3D CNNs. 2D features are extracted with the ImageNet pre-trained Resnet-152 [14] at the rate of one frame per second. 3D features are extracted with the Kinetics [4] pre-trained ResNeXt-101 16-frames model [12] to obtain 1.5 features per second. We aggregate features from longer video clips by the temporal max-pooling and concatenate 2D and 3D features to form a single 4096 dimensional vector for each video clip.

Text pre-processing. We preprocess transcribed video narrations by discarding common English stop-words. For the word representations, we use the GoogleNews pre-trained word2vec embedding model [34].

Training time. Once the video and text features are extracted, training our embedding model on the full HowTo100M dataset is relatively fast and takes less than three days on a single Tesla P100 GPU.

5.2. Datasets and evaluation setups

Action step localization. We evaluate localization of action steps in instructional videos on the recent CrossTask dataset [68]. CrossTask includes 18 tasks and 2.7k instructional videos with manually annotated action segments. Each video may contain multiple segments, corresponding to different actions. It also provides an ordered list of action steps with short natural language descriptions for each task. We apply our model trained only on HowTo100M to the problem of step localization by computing similarity between every frame in the video and the action label names of CrossTask. In order to compare to [68], we follow a similar inference procedure. We use the same recall metric as in [68], which is defined by the number of step assignments that fall into the correct ground truth interval, divided by the total number of steps in the video. Videos from the test set of CrossTask are removed from the HowTo100M training set to ensure that they are not observed at training time.

Text-based video retrieval. We also evaluate our learned embedding on the task of video clip retrieval using natural language queries. Given a textual description, the goal is to retrieve representative video clips from a large pool of videos. We evaluate our learned embedding using the standard recall metrics R@1, R@5, R@10 and the median rank (Median R). We provide experimental results for the following domain-specific video description datasets.

YouCook2 [67] is a cooking video dataset collected from YouTube. It features 89 different recipes and 14k video clips all annotated with textual descriptions collected from paid human workers. Since no descriptions are provided for the test set clips, we evaluate YouCook2 clip retrieval task on the validation clips (3.5k in total). Note that we have taken care to remove the few validation YouCook2 videos that are also present in HowTo100M.

MSR-VTT [58] is a dataset of generic videos collected from 257 popular video queries depicting 20 categories (including music, sports, movie, *etc*.) from YouTube. It contains 200k unique video clip-caption pairs, all annotated by paid human workers. We evaluate our model on the MSR-VTT clip retrieval test set used in [63] as performance of several other methods is reported on it.

LSMDC [44] is a dataset of movie clips. It features 101k unique video clip-caption pairs. All clips are associated with a description that either comes from the movie script or the audio description. We evaluate our model on the official LSMDC test set[2] that contains 1000 video-caption pairs.

5.3. Study of negative pair sampling strategy

We first study the effect of alternative strategies for sampling negative caption-video clip pairs when training our embedding. Table 3 shows that using negatives from the same video (intra-negatives) is beneficial as compared to randomly sampling them from other YouTube videos. The improvement is particularly significant on YouCook2 and CrossTask which are more fine-grained datasets than MSR-VTT and LSMDC. For the rest of the paper, we report numbers using our model trained with the intra-negative sampling strategy.

[2]https://sites.google.com/site/describingmovies/lsmdc-2016/movieretrieval

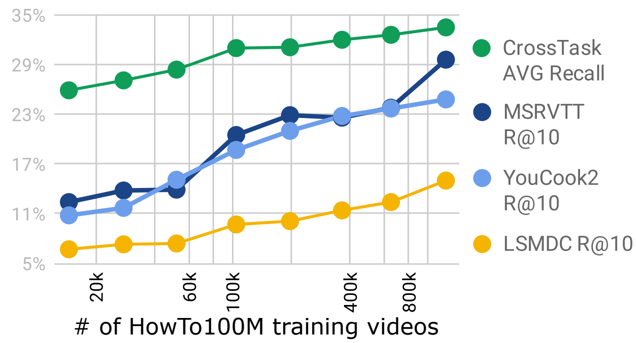

Figure 3: Retrieval and step localization results when varying the training size of our HowTo100M dataset.

5.4. Scale matters

A natural question is whether the large scale of our dataset is truly required to achieve high performance. To answer this, we train our embedding model on smaller subsets of our dataset. These smaller subsets of HowTo100M are created by gradually decreasing the allowed Youtube search rank (see the paragraph on data collection in Section 3.1 for more details) for training videos. We experiment with the following rank thresholds: top 2 (15k videos), top 3 (28k videos), top 5 (52k videos), top 10 (104k videos), top 20 (197k videos), top 40 (364k videos), top 80 (648k videos) and top 200 (entire HowTo100M dataset). This process ensures that we subsample training videos that are more likely to be relevant to the queried task as we reduce the size of the training dataset. Figure 3 shows average recall on CrossTask and the R@10 clip retrieval results on LSMDC, MSR-VTT and YouCook2 when varying the size of the training dataset. There is a clear improvement over all evaluated tasks with the gradual increase in the amount of training data. Interestingly, we do not observe any saturation, hence we can expect further improvements by collecting even more readily-available and unlabeled video data.

5.5. Comparison with state-of-the-art

CrossTask. We compare our off-the-shelf embedding trained on HowTo100M against methods proposed by Alayrac *et al*. [2] and Zhukov *et al*. [68] which is the current state-of-the-art on CrossTask for weakly supervised methods. Note that Zhukov *et al*. [68] have access to the ordered list of action labels at the task level and narrations are the only form of supervision during training. We also report the fully-supervised upper-bound from [68] obtained with a model that has been trained on action segments with ground truth annotation. The results are shown in Table 4. Our ap-

	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
Fully-supervised upper-bound [68]	19.1	25.3	38.0	37.5	25.7	28.2	54.3	25.8	18.3	31.2	47.7	12.0	39.5	23.4	30.9	41.1	53.4	17.3	31.6
Alayrac *et al*. [2]	15.6	10.6	7.5	14.2	9.3	11.8	17.3	13.1	6.4	12.9	27.2	9.2	15.7	8.6	16.3	13.0	23.2	7.4	13.3
Zhukov *et al*. [68]	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	**43.0**	13.3	22.4
Ours trained on HowTo100M only	**33.5**	**27.1**	**36.6**	**37.9**	**24.1**	**35.6**	**32.7**	**35.1**	**30.7**	**28.5**	**43.2**	**19.8**	**34.7**	**33.6**	**40.4**	**41.6**	41.9	**27.4**	**33.6**

Table 4: Step localization results on CrossTask [68] instructional video dataset.

Method	Trainset	R@1	R@5	R@10	Median R
Random	None	0.03	0.15	0.3	1675
HGLMM FV CCA [25]	YouCook2	4.6	14.3	21.6	75
Ours	YouCook2	4.2	13.7	21.5	65
Ours	HowTo100M	6.1	17.3	24.8	46
Ours	PT: HowTo100M FT: YouCook2	**8.2**	**24.5**	**35.3**	**24**

Table 5: YouCook2 clip retrieval results. PT denotes: pre-trained, while FT denotes: fine-tuned.

Method	Trainset	R@1	R@5	R@10	Median R
Random	None	0.1	0.5	1.0	500
C+LSTM+SA+FC7 [53]	MSR-VTT	4.2	12.9	19.9	55
VSE-LSTM [24]	MSR-VTT	3.8	12.7	17.1	66
SNUVL [64]	MSR-VTT	3.5	15.9	23.8	44
Kaufman *et al*. [22]	MSR-VTT	4.7	16.6	24.1	41
CT-SAN [65]	MSR-VTT	4.4	16.6	22.3	35
JSFusion [63]	MSR-VTT	10.2	31.2	43.2	13
Ours	HowTo100M	7.5	21.2	29.6	38
Ours	MSR-VTT	12.1	35.0	48.0	12
Ours	PT: HowTo100M FT: MSR-VTT	**14.9**	**40.2**	**52.8**	**9**

Table 6: MSR-VTT clip retrieval results. PT denotes: pre-trained, while FT denotes: fine-tuned.

proach significantly outperforms the state-of-the-art, even though it has not been specifically designed for the task of step localization in videos. The improvement made by our method is consistent across all tasks (with the exception of *Make Meringue*), showing that the trained model is not biased towards any specific domain. The recall is above 30% for most tasks with the significant improvement observed for the "*Add Oil to a Car*" task (6.4% to 30.7% boost in recall). Note that our method also outperforms the fully-supervised upper bound [68] on average. Thus, we conclude that training on a large amount of narrated videos is better than training a step localization model on a small but carefully annotated training set.

YouCook2 [67] does not provide an official benchmark nor any reported number for clip retrieval. As a consequence, we have applied a state-of-the-art text-video embedding model from Klein *et al*. [25] (HGLMM FV CCA) on YouCook2 using our features. We also report results of our model trained on YouCook2 instead of HowTo100M in Table 5. First, we notice that our off-the-shelf model trained on HowTo100M significantly outperforms both the exact same model directly trained on YouCook2 and [25]. Furthermore, fine-tuning our model pre-trained on HowTo100M on YouCook2 results in a significant improvement of 13.7 % in R@10 against [25]. In conclusion, we show that the off-the-shelf HowTo100M trained model can outperform state-of-the-art on this domain specific instructional video dataset. Moreover, we demonstrate that our model can get further benefits from fine-tuning.

MSR-VTT. We compare our model trained on **(i)** HowTo100M only, **(ii)** MSR-VTT only and **(iii)** pre-trained on HowTo100M and then fine-tuned on MSR-VTT against prior work that directly uses MSR-VTT for training (reproduced in [63]) in Table 6. Our off-the-shelf HowTo100M model outperforms [22, 24, 53, 64, 65] that are *directly trained* on MSR-VTT. Here again, after fine-tuning the HowTo100M pre-trained model on MSR-VTT, we observe a significant improvement over the state-of-the-art JSFusion [63] trained on MSR-VTT. However, as opposed to instructional videos (CrossTask) and cooking videos (YouCook2), training our model directly on MSR-VTT performs better than our off-the-shelf model trained on HowTo100M. We believe this is due to MSR-VTT videos being generic Youtube videos that are different from the instructional or VLOG type of videos that dominate HowTo100M. In Figure 4, we also investigate the impact on performance at various amounts of supervision when fine-tuning our pre-trained model. It shows that state-of-the-art performance [63] can be attained with *only* 20% of MSR-VTT samples. This has great practical implications as comparable performance can be obtained using significantly reduced annotation.

LSMDC. Finally, we compare to state-of-the-art on LSMDC in Table 7. This dataset is even more challenging as movie clips are quite distinct from HowTo100M videos. We compare against several other prior works that have been reproduced in [63] and are trained directly on LSMDC. Here again, we see that pre-training our model on HowTo100M and fine-tuning it on LSMDC also provides improvements upon a model directly trained on LSMDC.

Method	Trainset	R@1	R@5	R@10	Median R
Random	None	0.1	0.5	1.0	500
C+LSTM+SA+FC7 [53]	LSMDC	4.3	12.6	18.9	98
VSE-LSTM [24]	LSMDC	3.1	10.4	16.5	79
SNUVL [64]	LSMDC	3.6	14.7	23.9	50
Kaufman *et al.* [22]	LSMDC	4.7	15.9	23.4	64
CT-SAN [65]	LSMDC	4.5	14.1	20.9	67
JSFusion [63]	LSMDC	**9.1**	**21.2**	**34.1**	**36**
Ours	HowTo100M	4.0	9.8	14.0	137
Ours	LSMDC	**7.2**	18.3	25.0	44
Ours	PT: HowTo100M FT: LSMDC	7.1	**19.6**	**27.9**	**40**

Table 7: LSMDC clip retrieval results. PT denotes: pre-trained, while FT denotes: fine-tuned.

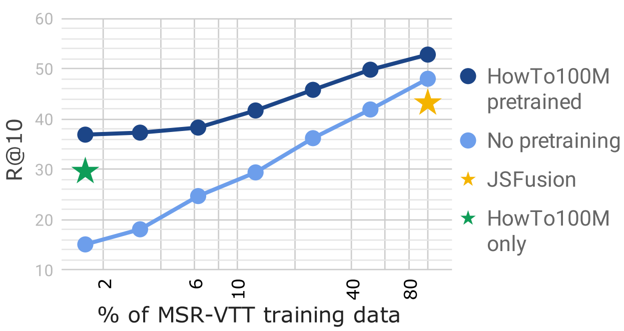

Figure 4: Evaluation of fine-tuning a HowTo100M pre-trained model with varying amounts of MSR-VTT supervision for text-to-video clip retrieval.

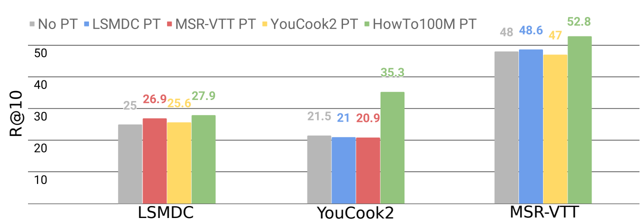

Figure 5: Results of clip retrieval by pre-training models on different datasets. Evaluation on LSMDC, YouCook2 and MSR-VTT.

This finding is interesting and shows that a HowTo100M pre-trained model can still be useful when fine-tuned on videos from a different domain.

5.6. Cross-dataset fine-tuning evaluation

In this section, we evaluate the advantage of HowTo100M for pre-training compared to pre-training on other smaller datasets. Figure 5 shows evaluation on YouCook2, MSR-VTT and LSMDC clip retrieval (R@10) using no pre-training (No PT), using pre-training on YouCook2, MSR-VTT, LSMDC and HowTo100M datasets while fine-tuning to the target dataset. For all evaluated datasets, pre-training on HowTo100M prior to fine-tuning on the target dataset consistently yields best results.

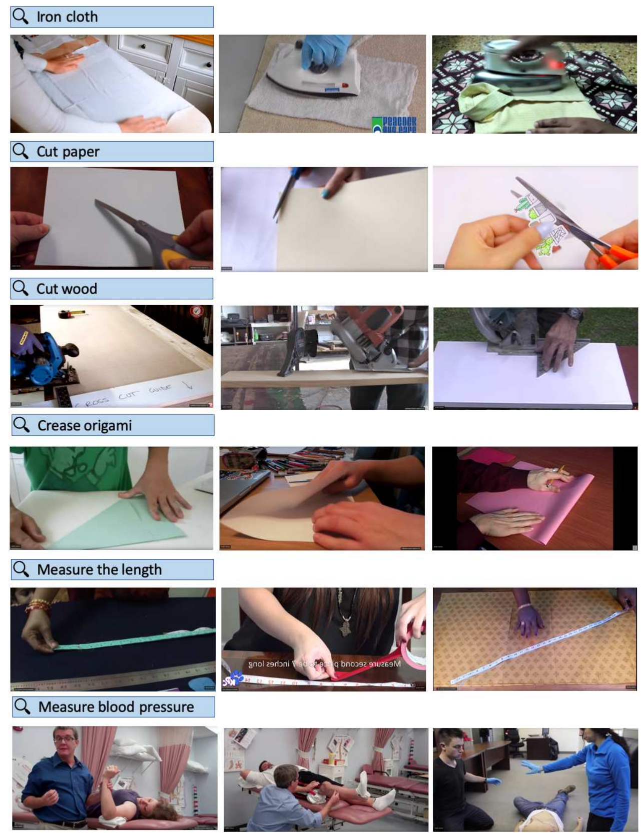

Figure 6: Example video-clip retrieval results on HowTo100M using our trained joint embedding.

5.7. Qualitative results

Figure 6 illustrates examples of retrieved video clips from HowTo100M using our trained joint text-video embedding. For example, our learned representation can correctly distinguish between queries *Cut paper* and *Cut wood*. A demo of the retrieval system is available online [1].

6. Conclusion

We have introduced HowTo100M, a video dataset with more than 130M video clips, extracted from 1.2M narrated web videos of people performing complex visual tasks. Our data collection method is fast, scalable and *does not require any manual annotation*. We use this dataset to learn a joint text-video embedding by leveraging more than 130M video clip-caption pairs. We have shown through various experiments that our learned embedding can perform better compared to models trained on existing carefully annotated but smaller video description datasets. The dataset, pre-trained models and code are available at [1].

Acknowledgements. The project was partially supported by Antoine Miech Google PhD fellowship, the MSR-Inria joint lab, the Louis Vuitton - ENS Chair on Artificial Intelligence, the ERC grant LEAP (No. 336845), the CIFAR Learning in Machines&Brains program, and the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468).

References

[1] Project webpage. https://www.di.ens.fr/willow/research/howto100m/, 2019. 1, 8

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 2, 6, 7

[3] Jean-Baptiste Alayrac, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *ICCV*, 2017. 2

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 5

[5] Kai Chen, Hang Song, Chen Change Loy, and Dahua Lin. Discover and learn new objects from documentaries. In *CVPR*, 2017. 2

[6] Mithun Chowdhury, Panda Rameswar, Evangelos Papalexakis, and Amit Roy-Chowdhury. Webly supervised joint embedding for cross-modal image-text retrieval. In *ACM International Conference on Multimedia*, 2018. 2

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 2

[8] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019. 2

[9] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pages 457–468, 2016. 2

[10] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014. 2

[11] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014. 2

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018. 5

[13] David Harwath, Adrià Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*, 2018. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 5

[15] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. *ICCV*, 2017. 1, 2, 5, 12

[16] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016. 2

[17] De-An Huang, Joseph J. Lim, Li Fei-Fei, and Juan Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. In *CVPR*, 2017. 2

[18] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. Finding "it": Weakly-supervised reference-aware visual grounding in instructional video. In *CVPR*, 2018. 2

[19] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *preprint*, 2018. 3

[20] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016. 2

[21] Andrej Karpathy, Armand Joulin, and Fei Fei F Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014. 5

[22] Dotan Kauman, Gil Levi, Tal Hassner, and Lior Wolf. Temporal tessellation: A unified approach for video analysis. In *ICCV*, 2017. 7, 8

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[24] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2014. 7, 8

[25] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 1, 2, 7

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

[27] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*, 2016. 2

[28] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 3

[29] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 2

[30] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *NAACL*, 2015. 2

[31] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 5

[32] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *arXiv:1804.02516*, 2018. 1, 2, 4, 5

[33] 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. *arXiv preprint arXiv:1906.03327*, 2019. 4, 5

[34] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 5

[35] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, 2018. 2

[36] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016. 1, 2

[37] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 1, 2

[38] Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *CVPR*, 2017. 1, 2

[39] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understandingby Generative Pre-Training. *preprint*, 2018. 3

[40] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *preprint*, 2019. 3

[41] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017. 2

[42] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *CVPR*, 2018. 2

[43] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 2

[44] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 2017. 2, 5, 6

[45] Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS, 2018. 2

[46] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *CVPR*, 2018. 2

[47] Ozan Sener, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[48] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016. 2

[49] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 3

[50] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 2

[51] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 1, 2

[52] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015. 2

[53] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016. 7, 8

[54] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *PAMI*, 2018. 1, 2, 5

[55] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, pages 5005–5013, 2016. 1, 2, 5

[56] Xin Wang, Jiawei Wu, Da Zhang, Yu Su, and William Yang Wang. Learning to compose topic-aware mixture of experts for zero-shot video captioning. In *AAAI*, 2018. 2

[57] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *ICCV*, 2017. 2

[58] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1, 2, 5, 6

[59] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, volume 5, page 6, 2015. 1, 2

[60] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, pages 4651–4659, 2016. 2

[61] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, pages 4584–4593, 2016. 1, 2

[62] Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *ACM*, 2014. 2

[63] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 1, 2, 6, 7, 8

[64] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. Video captioning and retrieval models with semantic attention. In *ECCV LSMDC2016 Workshop*, 2016. 5, 7, 8

[65] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017. 7, 8

[66] Luowei Zhou, Xu Chenliang, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2

[67] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2, 5, 6, 7

[68] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 2, 5, 6, 7