Video-Text Representation Learning

Yash Sharma
Rutgers University
yss26@rutgers.edu

Monaal Sanghvi
Rutgers University
ms3185@rutgers.edu

Jenil Jain
Rutgers University
jj822@rutgers.edu

Akshay Patil
Rutgers University
avp119@rutgers.edu

Abstract

The concept of cross-modal retrieval has found itself steadily increasing in usage and popularity because of its ability to handle and analyze multi-sensor data. Many real-world scenarios consist of instances where data is collected from several sensors like images, videos, audio, etc. As we start dealing with multi-modal data increasingly, it is becoming important perform cross modal retrieval to leverage and represent such form of data and handle different modalities. However, real-world data like video and text data include changing contexts and different semantics when broken down into frames or words. In such cases, context in videos, or sentences and semantics in a paragraph can change frequently. To overcome this challenge, we utilize a Cooperative Hierarchical Transformer (COOT)[4] to extract information from hierarchical components of video like frames and clips, and hierarchical components of text like sentences and words and map different modalities to perform the task of video-text retrieval. We evaluate performance using recall and median rank for each hierarchy to determine the best possible hierarchy level for retrieval. The evaluation at sentence-clip level serves the step localisation part.

Keywords: *Video, text, cross-modal, hierarchy, modalities*

1. Introduction

Video-text retrieval is a critical component of multi-modal research and is widely employed in real-world web applications. This field of study has lately exploded, thanks to the success of self-attention in text analysis and its near-instant cross-modal applicability.

In this work, we produce a hierarchical model that can exploit the long range temporal context both in videos and texts while learning joint cross modality. We use the concept of a hierarchical model with losses to capture the entire temporal environment.These losses enforce the interaction within and across different hierarchy levels. There are three degrees of hierarchy taken into consideration: frame/word, clip/sentence, and video/paragraph. We also incorporate alignment losses from Zhang et al. [9] and augment our baseline model with a novel feature aggregation approach for intra-level feature interactions and a new transformer-based module for inter-level interactions (between local and global semantics)

2. Motivation

There are several methods of performing video-text representation. We have made use of Cooperative Hierarchical Transformer (COOT). The idea behind the selection of a transformer based learning technique is because transformers have been greatly successful for NLP and CV downstream tasks as most of the learning happens directly through data with minimal inductive biases. What transpired us to use the Cooperative Hierarchical Transformer is it uses different hierarchies in text and video to generate best possible representation. A hierarchy in a video-text context represents various components of the data. For a video, the hierarchy starts from the entire video or recipe, then the short clips which are formed from videos, and then the frames which are individual images in each clip.Thus the hierarchical order is recipe video, clips and frames. For text, the hierarchical order is paragraphs, then sentences which are segregated from paragraphs and words which are segregated from sentences. This hierarchical analysis allows us to learn local context between these clips as well as the global context.

Another advantage of COOT is that while other multi-modal transformers like MDMMT use too many pre-trained expert models for feature extraction, COOT follows a much simpler architecture wherein we only need video features from a model pretrained on HowTo100M [5]and text features from BERT.

Also since COOT follows a simpler architecture, the model has only 10.6 million parameters which greatly reduces the training time to 3 hours on two GTX 1080 Ti GPUs.

3. Prior Work

Previous work of video-text representation involves methods such as MIL-NCE[5] which are also models that extract features via training on HowTo100M dataset. However, this model does not utilize hierarchical information and cannot outperform COOT in terms of recall rate and median rank. For video-paragraph hierarchy, MINCE performs worse than COOT with a 16.4% R@1 score.

The Multidomain Multimodal Transformer for Video Retrieval (MDMMT)[3] model performs video-rext retrieval using MSRVTT dataset[8] dataset and LSMDC dataset. The model with minimal finetuning produces good results for three different benchmarks i.e. MSRVTT, LSDMDC and ActivityNet. The model outperforms some SOTA methods but does not improve upon the hierarchical architecture model used by COOT. It also produces different approaches for generating the best possible overlap between training and testing datasets. The model obtains 0.201, 0.451, 0.580 values for recall at 1, 5, and 10 respectively and a value of 7.0 for median rank.

4. Technical Details

4.1. Major Components

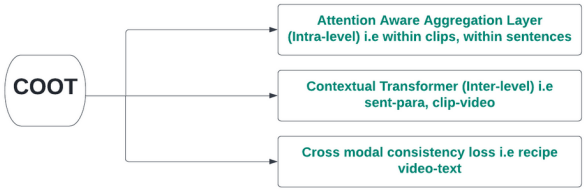

Figure 1. COOT major components

The COOT architecture consists of three major components:

1. An attention-aware feature aggregation layer, which leverages the local temporal context for example within a clip or sentence i.e. intra-level context.
2. A contextual transformer to learn the interactions between low-level and high-level semantics for example between clip-video, sentence-paragraph i.e. inter-level context.
3. A Cross-modal cycle-consistency loss to connect video and text.

4.2. Model Architecture

The model consists of two branches: one for video input (top) and one for text input (bottom). Both the modalities i.e video and text follow the same architecture. We encode frame-level/word-level information from a video and its related text. To extract clip/sentence-level features, features from each segment (clip/sentence) are passed through a standard temporal transformer (T-Transformer) followed by the suggested feature aggregation module (Attention-FA). Finally, depending on interactions between local context (clip/sentence characteristics) and global context (all frames/words features), a new contextual transformer generates the final video/paragraph embedding. $\ell^{L}_{\text{align}}, \ell^{H}_{\text{align}}$, ℓ^{g}_{align} and ℓ_{CMC} enforce the model to align the representations at different levels.

Attention-aware aggregation layer models the intra-level computation, which focuses on the temporal interactions dedicated to low-level entity.

We develop a contextual attention module for inter-level collaboration, which forces the network to highlight semantics relevant to the video's overall context while suppressing unrelated semantics. This is accomplished by simulating the interaction of low-level and high-level entities.

A new cross-modal cycle-consistency loss is also used to impose interaction between modalities and foster semantic alignment in the learnt common space. In addition to these architectural contributions, constraining two domains to develop consistent representations improves semantic alignment significantly.

4.3. Intra-level Cooperation

An attention-aware feature aggregation layer is used to focus on temporal interactions between low-level entities to model intra-level cooperation[6]. For example, consider a video of someone cooking, objects placed on the table are more significant compared to those in the background. As a result, depending on the situation, we must pay attention to specific characteristics.

Consider a sequence of T feature vectors, represented by $X = \{x_1, \ldots, x_T\}$ (e.g. $\hat{f}^{k}_{i,:} = \left\{\hat{f}^{k}_{i,1}, \ldots, \hat{f}^{k}_{i,T}\right\}$). We set key $K = X$ and utilize two learnable transformation weights W_2 and W_1 together with two biases b_1 and b_2. The attention matrix A is computed as:

$$A = \text{softmax}\left(W_2 Q + b_2\right)^T,$$
$$Q = GELU\left(W_1 K^T + b_1\right), K = X$$

The final feature is computed as $\hat{x} = \sum_{i=1}^{T} a_i \odot x_i$, where $\odot$ denotes element-wise multiplication and a_i is the i-th attention vector of A for the i-th feature. We feed $\left\{\hat{f}^{k}_{i,:}\right\}_{i=1}^{n}$ and $\hat{f}^{k}_{:}$ to this component and obtain the clip-level $\left(\left\{\vartheta^{k}_{i}\right\}_{i=1}^{n}\right)$ features and the global context for the video (g_ν).

4.4. Inter-level Cooperation

A contextual attention module is developed for inter-level collaboration, which forces the network to highlight

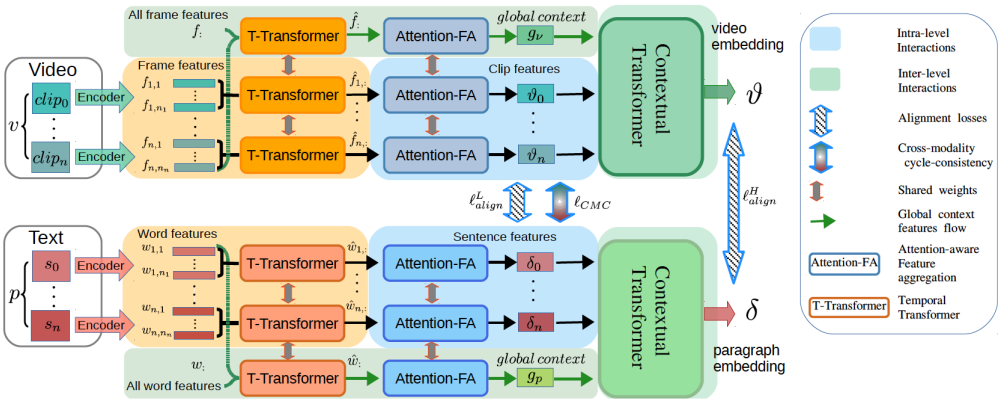

Figure 2. COOT Architecture

semantics relevant to the video's overall context while suppressing unrelated semantics. Using this, the network learns to put more focus on semantics relevant to the video's overall context and discard irrelevant ones by modeling the relationships between local and global context.

Thus, a Contextual Transformer (CoT) in Figure 2-Right is used to represent how low-level and high-level semantics interact. The Contextual Transformer is build with two modules F_{Local} and F_{Global}. The positional embedding is added to the inputs of F_{Local}. The goal of F_{Local} aims to simulate low-level semantic exchanges. $\left(\{\vartheta_i^k\}_{i=1}^n\right)$, whereas F_{Global} models the interactions between local and global context (g_ν) to highlight the important semantics.

For these local representations $\{\vartheta_i^k\}_{i=1}^n \in \mathbb{R}^{n\times d}$, where n is the number of clips and d indicates the feature dimension, F_{Local} a multi-head attention followed by a feed-forward layer and a normalization layer is applied on top of both layers which produces embeddings $\{h_i\}_{i=1}^n$.

Key (K)-value(V) pairs are computed on these embeddings $\{h_i\}_{i=1}^n \in \mathbb{R}^{n\times d}$ and query(Q) based on the global context g_v.F_{Global} produces the attention output as follows,

$$H_{attn} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \quad Q = \mathcal{W}_q g_v,$$
$$K = \mathcal{W}_k \{h_i\}_{i=1}^n, V = \mathcal{W}_v \{h_i\}_{i=1}^n$$

where $\mathcal{W}_q, \mathcal{W}_k$, and $\mathcal{W}_v$ are the embedding weights. H_{attn} is a weighted sum of values (local semantics), where the weight of each value is calculated based on its interaction with the global context query Q. H_{attn} is further encoded by a feed-forward layer to produce the contextual embedding H_{context} . We calculate the mean of $\{h_i\}_{i=1}^n$ and concatenate it with H_{context} to obtain the final video embedding $\vartheta^k = \text{concat}\left(\text{mean}\left(\{h_i\}_{i=1}^n\right), H_{\text{context}}\right)$

4.5. Local and Global context

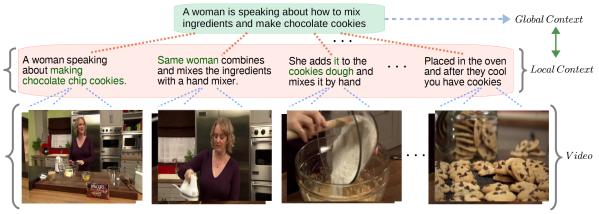

Figure 3. Local and Global context

Before we go deeper into the working of each stage, the terms local and global context are vital to understand. Local context models short term interactions between low level semantics whereas global context models interactions between local and global semantics. For a model to perform better, the global context plays an important part along with the local context. For example, in Fig.3. it can be seen that a lady is making chocolate cookies, the model must learn from general or global context the type of flour that will be used in general baking recipes. Also when the sentence says the same woman i.e. a local context, the model must be aware of the person's identity. As a result, the model is encouraged to maximize representations in terms of interactions between local and global context.

4.6. Loss Functions

Semantic Alignment Losses:

We have defined semantic alignment losses for each hierarchy level i.e. word-frame, clip-sentence, recipe-paragraph. Semantic alignment losses make use of contrastive loss that ensures positive samples stay close to each other and negative samples are pushed farther apart from each other. Hence, we maximize the distance between a negative video sample and a positive text. On top of

this, we add the maximum distance between positive video sample and negative text sample. We calculate these losses for all levels of hierarchy. The general equation of the semantic alignment loss function is given as follows:

$$L(\mathcal{P},\mathcal{N},\alpha) = \max\left(0,\alpha + D(x,y) - D\left(x',y\right)\right) + \max\left(0,\alpha + D(x,y) - D\left(x,y'\right)\right)$$

where,

α = margin,
$\mathcal{P}$ = positive sample pair, denoted by $\mathcal{P} = (x,y)$
$\mathcal{N}$ = Negative sample pair, denoted by $\mathcal{P}= (x\prime,y),(x,y\prime)$

Low-level semantic alignment: Given a clip-sentence $\left(\vartheta_i^k,\delta_i^k\right)$, the low level semantic loss is defined as follows:

$$\ell_{\text{align}}^{L} = \sum_{k\in\mathcal{D},i,k'\neq k,i'\neq i} L\left(\left(\vartheta_i^k,\delta_i^k\right), \left\{\left(\vartheta_i^k,\delta_{i'}^{k'}\right),\left(\vartheta_{i'}^{k'},\delta_i^k\right)\right\},\beta\right)$$

High Level semantic alignment: Given a video-paragraph, $\left(\left(\vartheta^k,\delta^k\right)\right.$, the high level semantic loss is defined as follows:

$$\ell_{align}^{H} = \sum_{k\in\mathcal{D},k'\neq k} L\left(\left(\vartheta^k,\delta^k\right),\left\{\left(\vartheta^k,\delta^{k'}\right),\left(\vartheta^{k'},\delta^k\right)\right\},\alpha\right)$$

Global semantic alignment: Given a global context (g_v,g_p), the global semantic loss is defined as follows:

$$\ell_{\text{align}}^{g} = \sum_{k\in\mathcal{D},k'\neq k} L\left(\left(g_v^k,g_p^k\right),\left\{\left(g_v^k,g_p^{k'}\right),\left(g_v^{k'},g_p^k\right)\right\},\alpha_g\right)$$

Cluster Loss:

The cluster loss is defined as the loss that aligns low level and high level hierarchies. Cluster loss also pushes appear positive and negative sample embeddings. It uses a $(1,1)$ parameter are pair used to denote the non-changing positive embeddings.

The equation for cluster loss is given as follows:

$$\ell_{cluster} = \sum_{k\in\mathcal{D},i,k'\neq k,i'\neq i} L\left((1,1),\left\{\left(\vartheta_i^k,\vartheta_{i'}^{k'}\right),\left(\delta_{i'}^{k'},\delta_i^k\right)\right\},\gamma\right) + \sum_{k\in\mathcal{D},k'\neq k} L\left((1,1),\left\{\left(\vartheta^k,\vartheta^{k'}\right),\left(\delta^{k'},\delta^k\right)\right\},\eta\right)$$

Cross-modal cycle-consistency Loss:

Cross-modal cycle consistency[2] works as follows:
For a sentence i, find its nearest video clip semantically. For this video clip, find its semantically nearest sentence j. For this sentence j, find its semantically nearest sentence. The nearest neighbor for j should be the sentence i.
The embedding is said to be semantically consistent only if it points back to the original location. Cycle-consistency[2, 7] penalizes deviations that have deviated and thus minimizes the value of the loss function.

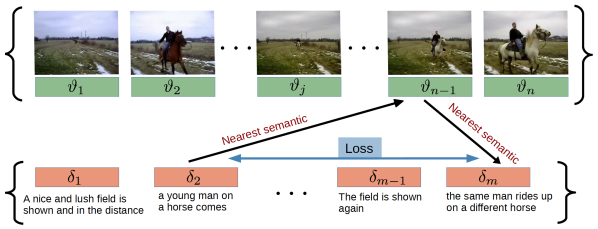

Figure 4. Image to Recipes Median Rank

$$\bar{\vartheta}_{\delta_i} = \textstyle\sum_{j=1}^{n} \alpha_j \vartheta_j$$

$$\text{where} \quad \alpha_j = \frac{\exp\left(-\|\delta_i - \vartheta_j\|^2\right)}{\sum_{k=1}^{n}\exp\left(-\|\delta_i - v_k\|^2\right)}$$

Hence $\{\vartheta_i\}_{i=1}^{n}.\alpha_j$ is the similarity score of clip ϑ_j to sentence δ_i. We sentence sequence $\{\delta_i\}_{i=1}^{m}$ and calculate the soft location

$$\mu = \sum_{j=1}^{m} \beta_j j$$

$$\text{where} \quad \beta_j = \frac{\exp\left(-\left\|\bar{\vartheta} - \delta_j\right\|^2\right)}{\sum_{k=1}^{m}\exp\left(-\left\|\bar{\vartheta} - \delta_k\right\|^2\right)}$$

$$\ell_{CMC} = \|i - \mu\|^2$$

Final Loss: The Final loss function(ℓ_{final}) is calculated as the sum of all the three alignment losses ($\ell_{\text{align}}^{L}, \ell_{align}^{H}, \ell_{\text{align}}^{g}$), cluster loss ($\ell_{cluster}$), and cross-modal cycle-consistency loss (ℓ_{CMC}).

The equation of the final loss function ℓ_{final} is given as follows:

$$\ell_{final} = \ell^{L}_{\text{align}} + \ell^{H}_{align} + \ell^{g}_{\text{align}} + \ell_{cluster} + \lambda \ell_{CMC}$$

5. Evaluation

5.1. Dataset

We have used the YouCook2[10] video dataset which contains 2000 long untrimmed videos and 89 cooking recipes. Each video contains procedure steps in annotated with temporal boundaries. Each procedure step is also described in sentences. The dataset is separated into train, test and validation text files which contain a list of videos for each set. The model evaluation is performed using Median Rank and Recall Rate evaluation metrics. The median rank and Recall rate is calculated for average concatenated embeddings, title embeddings, instructions embeddings and ingredients embeddings.

5.2. Evaluation Metrics

Recall: Recall is defined as the number of steps correctly assigned to the correct ground truth time interval divided by the total number of steps and is the least strict metric out of the three considered.

Median Rank: We first calculate the rank of recipe embeddings for a subset of image embeddings. We do this for a few sets of random images. Then we take the median of these ranks which will give us the median rank.

6. Discussion of results

	R@1	R@5	R@10	R@50	MedR
Vid-Par	0.810	0.958	0.978	0.996	2.2
Par-Vid	0.783	0.963	0.978	0.996	2.3
Clip-Sen	**0.159**	**0.395**	**0.512**	**0.782**	**74.4**
Sen-Clip	**0.169**	**0.406**	**0.525**	**0.780**	**73.2**

Table 1. Results on test set of 457 videos

From the results obtained, it can be observed that recall and median rank improve with increasing levels of hierarchy.

6.1. Ablation Studies

To observe changes in model's performance, we perfomed ablation studies by keeping all the loss functions to 0 one by one.

	R@1	R@5	R@10	R@50	MedR
Vid-Par	0.720	0.950	0.965	0.993	2.6
Par-Vid	0.705	0.952	0.969	0.993	2.7
Clip-Sen	**0.042**	**0.117**	**0.176**	**0.352**	**636.1**
Sen-Clip	**0.035**	**0.104**	**0.156**	**0.313**	**706.2**

Table 2. Results keeping ℓ^{L}_{align} to 0

From the ablation studies done by keeping ℓ^{L}_{align} to 0, we see a decrease in performance for R@1 for Clip-Sentence retrieval and Sentence-Clip retrieval.

	R@1	R@5	R@10	R@50	MedR
Vid-Par	0.486	0.705	0.768	0.891	20.6
Par-Vid	0.492	0.718	0.812	0.921	19.2
Clip-Sen	**0.155**	**0.377**	**0.509**	**0.765**	**84.9**
Sen-Clip	**0.166**	**0.390**	**0.521**	**0.771**	**83.5**

Table 3. Results keeping ℓ^{H}_{align} to 0

From the ablation studies done by keeping ℓ^{H}_{align} to 0, we see a decrease in performance for R@1 for Video-Paragraph retrieval and Paragraph-Video retrieval.

	R@1	R@5	R@10	R@50	MedR
Vid-Par	0.805	0.963	0.980	0.998	2.0
Par-Vid	0.772	0.956	0.974	0.996	2.4
Clip-Sen	**0.145**	**0.364**	**0.487**	**0.755**	**94.1**
Sen-Clip	**0.160**	**0.375**	**0.500**	**0.755**	**94.3**

Table 4. Results keeping ℓ_{CMC} to 0

From the ablation studies done by keeping ℓ_{CMC} to 0, we don't see any significant changes in the performance.

6.1.1 Video alignment

Figure 5. Boil the potatoes in water. Add chopped potatoes to the pan. Add butter and mash. Add some milk and mash.

7. Conclusion

A hierarchical cooperative transformer architecture is offered for learning a joint video and text embedding space with aligned semantics. The architecture is built to encourage the cross-level utilization of long-range temporal

context. To model interactions inside and across hierarchy levels, two novel components are used: an attention-aware feature aggregation module to model interactions between frames and words, and a contextual transformer to represent interactions between local contexts and global context. A new cross-modal cycle-consistency loss ensures that clips and phrases are semantically aligned. We have demonstrated that both components work together and individually to improve retrieval performance.

8. Contributions

VIDEO-TEXT REPRESENTTION LEARNING

Rutgers University
CS536: Machine Learning

TASK	ASSIGNED TO
Project Proposal	
Literature Review	All members
Proposal	All members
Project Stage 1	
Text Embeddings	Akshay, Monaal
Image Embeddings	Jenil, Yash
CCA Training	Akshay, Monaal
CCA Evaluation	Jenil, Yash
Project Stage 2	
Deep CCA Literature Review	All members
Deep CCA Training	Akshay, Monaal
Deep CCA Evaluation	Jenil, Yash
Final Stage	
Literature Review	All members
Video Embeddings Using HowTo100M	Jenil, Yash
Text Embeddings using BERT base uncsaed	Akshay, Monaal
COOT transformer	Jenil, Yash
Model Evaluation and result calculation	Akshay, Monaal
Final Report and Presentation	All members

Figure 6. Contributions

9. Other approaches

We also tried to replicate the Drop-DTW[1] loss with features from MIL-NCE. The code for which is linked in the project code section. We tried to fine tune the CLIP model for video retrieval.

10. Project Code

Please find the link for COOT implementation here: COOT

Please find the link for DropDTW implementation here:
DROP-DTW

References

[1] Mikita Dvornik, Isma Hadji, Konstantinos G Derpanis, Animesh Garg, and Allan Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers. *Advances in Neural Information Processing Systems*, 34, 2021. 6

[2] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 4

[3] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. Mdmmt: Multidomain multimodal transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3363, 2021. 2

[4] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning, 2020. 1

[5] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 1, 2

[6] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 2

[7] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 4

[8] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 5288–5296, 2016. 2

[9] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 374–390, 2018. 1

[10] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, pages 7590–7598, 2018. 5